DTU Library

D-NA5.1 Smart Grid configuration validation scenario description method

Heussen, Kai; Bondy, Daniel Esteban Morales; Nguyen, Van Hoa; Blank, Marita; Klingenberg, Thole; Kulmala, Anna; Abdulhadi , Ibrahim F. ; Pala, Daniele; Rossi, Marco; Carlini , Claudio

Total number of authors:
17

Publication date:
2017

Document Version
Publisher's PDF, also known as Version of record

European Research Infrastructure supporting Smart Grid Systems Technology Development, Validation and Roll Out

Work Package 05

NA5 - Holistic System Integration and Testing Procedure

Deliverable D5.1

D-NA5.1 Smart Grid configuration validation scenario description method

Grant Agreement No:	**654113**
Funding Instrument:	**Research and Innovation Actions (RIA) – Integrating Activity (IA)**
Funded under:	**INFRAIA-1-2014/2015: Integrating and opening existing national and regional research infrastructures of European interest**
Starting date of project:	**01.11.2015**
Project Duration:	**54 month**

Contractual delivery date:	**31.01.2017**
Actual delivery date:	**01.05.2017**
Name of lead beneficiary for this deliverable:	**6 Danmarks Tekniske Universitet**
Deliverable Type:	**Report (R)**
Security Class:	**Public (PU)**
Revision / Status:	**released**

Project co-funded by the European Commission within the H2020 Programme (2014-2020)

Document Information

Document Version:	6
Revision / Status:	released

All Authors/Partners

Kai Heussen / DTU
Daniel Esteban Morales Bondy / DTU
Van Hoa Nguyen / GINP
Marita Blank / OFF
Thole Klingenberg / OFF
Anna Kulmala / OFF
Ibrahim F. Abdulhadi / UST
Daniele Pala / RSE
Marco Rossi / RSE
Claudio Carlini / RSE
Arjen van der Meer / TUD
Panos Kotsampopoulous / ICCS
Alexandros Rigas / ICCS
Ata Khavari / DERLab
Quoc Tuan Tran / CEA
Cyndi Moyo / AIT
Thomas Strasser / AIT

Distribution List ERIGrid consortium members

Document History

Revision	Content / Changes	Resp. Partner	Date
1	Initial integration of the document	DTU	15.06.16
2	Integration of state of the art sections	DTU	07.09.16
3	Reverted to a skeleton/outline, with links to the appropriate documents of each working group	DTU	12.10.16
4	Configuration of template for final deliverable	DTU	10.01.17
5	Deliverable for Steering Committee Review	DTU	28.02.17
6	Final version	DTU	08.03.17

Document Approval

Final Approval	Name	Resp. Partner	Date
Review WP Level	Matthias Uslar	OFF	12.12.16
Review WP Level	Boye Annfelt Høverstad	SIN	24.02.17
Review WP Level	Julia Merino Fernández	TEC	28.02.17
Review Steering Com. Level	Thomas Strasser	AIT	01.05.17

Disclaimer

Table of contents

Abbreviations

DER	Distributed Energy Resource
C-AD	Connectivity - Abstract Data
C-DD	Connectivity - Direct Data
C-DP	Connectivity - Direct Physical
C-IP	Connectivity - Indirect Physical
CP	Connection Point (System Configuration)
C-S-D	Connectivity - Stakeholder - Directive
C-S-I	Connectivity - Stakeholder - Informational
C-S-O	Connectivity - Stakeholder - Owner
C-S-OP	Connectivity - Stakeholder - Operator
C-S-R	Connectivity - Stakeholder - Responsible
C-S-T	Connectivity - Stakeholder - Transactive
CTR-IP	Controllable Input Parameter
HIL	Hardware in the loop
P-HIL	Power hardware in the loop
C-HIL	Controller hardware in the loop
DoA	Description of Action
DoE	Design of Experiments
DTH	Domain Type Hierarchy
DuI	Domain under Investigation
E-SC	Experiment Specific System Configuration
FuI	Function(s) under Investigation
FuT	Function(s) under Test
GSC	Generic System Configuration
ICT	Information and Communication Technology
IP	Input Parameter
RI-GSC	Research Infrastructure Generic System configuration
RI-SC	Research Infrastructure System Configuration
OP	Output Parameter
OuI	Object(s) under investigation
PoI	Purpose of Investigation
SC	System Configuration
SC	Specific System Configuration
SCC	System Configuration Container
SCD	System Configuration Diagram
SCType	System Configuration Container Type
SuT	System under test
T	Terminal (System Configuration)

TC	Test Case
TC-GSC	Test Case Generic System Configuration
TM	Target Metrics
TS	Test System
TS-SC	Test Specific System Configuration
UC	Use Case
UC-GSC	Use Case Generic System Configuration
UC-IP	Uncontrollable Input Parameter

Executive Summary

Smart Grid solutions have become complex and multidisciplinary. With the further integration of ICT solutions and other energy systems new test processes must be defined. A method for framing a holistic approach to testing has been developed in order to capture this complexity, which aims at enabling the testing of new solutions within their relevant operational context. A holistic testing method ensures a clear vocabulary for Smart Grid testing across engineering disciplines and a common understanding of how to describe a testing approach that addresses a system-relevant perspective.

The holistic testing vision outlined for ERIGrid is widening the scope of conventional testing:

- Tequirements associated with multiple domains are viewed as part of single test case
- Systematic and integrated testing strategy for systems, components and their integration
- The hybridization of methods applicable to distinct formal representation frameworks (i.e. ICT, discrete & logic oriented testing, vs. physical continuous models and uncertainty)
- The formal integration of several independent tests into a common framework
- Technical integration of different means of testing, such as real-time simulation

The proposed holistic testing procedure aims to support the integration and alignment of these required aspects is outlined in Figure 0.1.

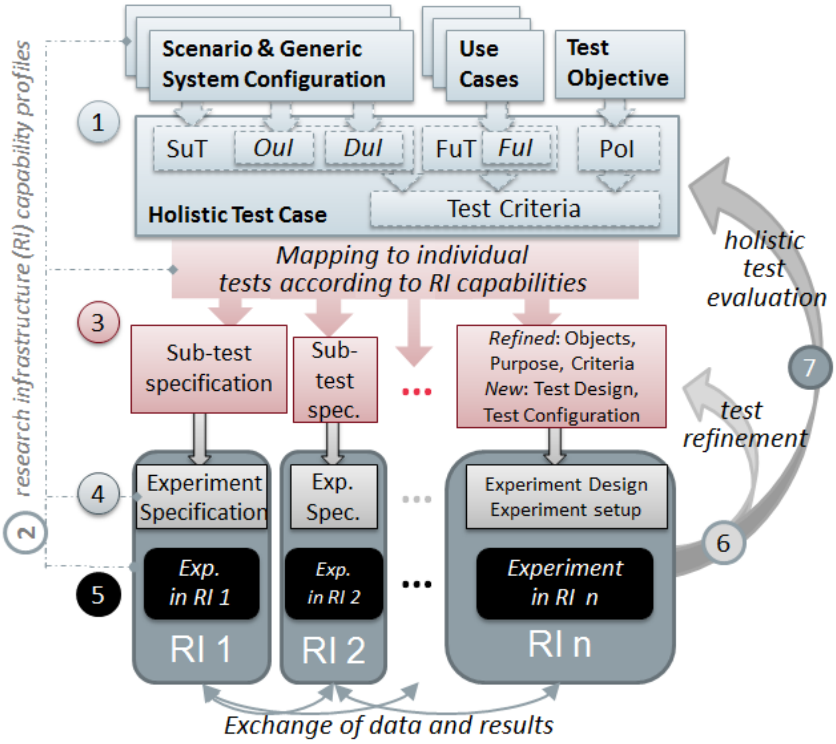


Figure 0.1: Main steps of the ERIGrid methodology applied to a 'holistic' test case, which then is divided into sub-tests to be performed at several laboratories

The sequence of specifications includes:

1. **Test case** – in analogy to 'use case': defining the objectives and domains for a test
2. **Test specification** – what test is to be carried out? defining test system and its parameters
3. **Experiment specification** – how the test specification is to be implemented in a given Research Infrastructure (RI)

The (holistic) test case, which describes in the test objective and context a specific Smart Grid solution, is refined test specifications that can be split among different RIs, yet still maintaining the overall test problem. Finally, these RIs must then transform these test specifications into experiments specifications that can be carried out at the individual RI. The practical use of the approach is supported by concise definitions, template forms with guiding text, graphical templates as well as exemplary applications.

To support multi-domain approach, a domain-independent system configuration description method has been defined. This method can be applied to system configurations in several contexts, so that for each test description template a related system configuration type is defined (TC-GSC, TS-SC, E-SC) in Table 0.1. Three further contexts of system configuration specifications have been identified (UC-GSC, RI-SC, and RI-GSC) as reported below.

Table 0.1 Classification of System Configuration Types

Name/ Purpose	Context / Document	GSC/ (S)SC	SCType	Explanation
Function-System Alignment	Use Case	GSC	UC-GSC	As SGAM domains & zones: reference designation for functions, independent of test case. Corresponds to D-JRA1.1 Generic System Configurations.
Test Case context model	Test Case	GSC	TC-GSC	Establishes type conventions for test case: relevant SC component types, domains, etc., and categorically identifies the SuT (and optional OuIs); specifies multiplicities; “class model”.
Test System	Test Specification	(S)SC	TS-SC	A concrete instance of TC-GSC to address a specific OuI and test criteria; labelled terminals and specific connections; OuI and SuT identified as overlay annotation.
Experiment Setup	Experiment Specification	(S)SC	E-SC	The configuration and interconnection of RI components, representing the SuT, and including OuI; also “Test Setup”
RI Description	RI database entry	(S)SC	RI-SC	Lab configuration with components, including potential multiplicity and potential connectivity of lab components, but may have undefined connectivity.
RI information model	RI profiling	GSC	RI-GSC	Specification of Lab profiling data structures, including component types and domain types.

Here, GSC refers to a Generic System Configuration, and (S)SC to a Specific System Configuration. While the GSCs define a contextual information model, a (S)SC defines a specific instance of a system configuration.

A visual representation of how the three test case specification layers in terms of associated system configurations (TC-GSC, TS-SC, and E-SC) connect is shown in Figure 0.1 and Figure 0.2. In this exemplary case, an inverter connected to a distribution system is controlled by a centralized voltage controller. To test an inverter (here the Object under Investigation (OuI) in closed loop with the voltage controller, an operational context has to be identified, called the System under Test (SuT).

In the TC-GSC, the SuT generic elements and context are identified, which is then refined to a concrete test system in the TS-SC. The TS-SC elements are then mapped into a laboratory context in the E-SC, where in this case only the OuI (the inverter) is found as a physical component, and the remaining components for the TS-SC are emulated.

Addressing the requirements of a testing process, the proposed methodology attempts to strike a balance between formal definitions, existing concepts within standards, and the practical use and understanding of tests.

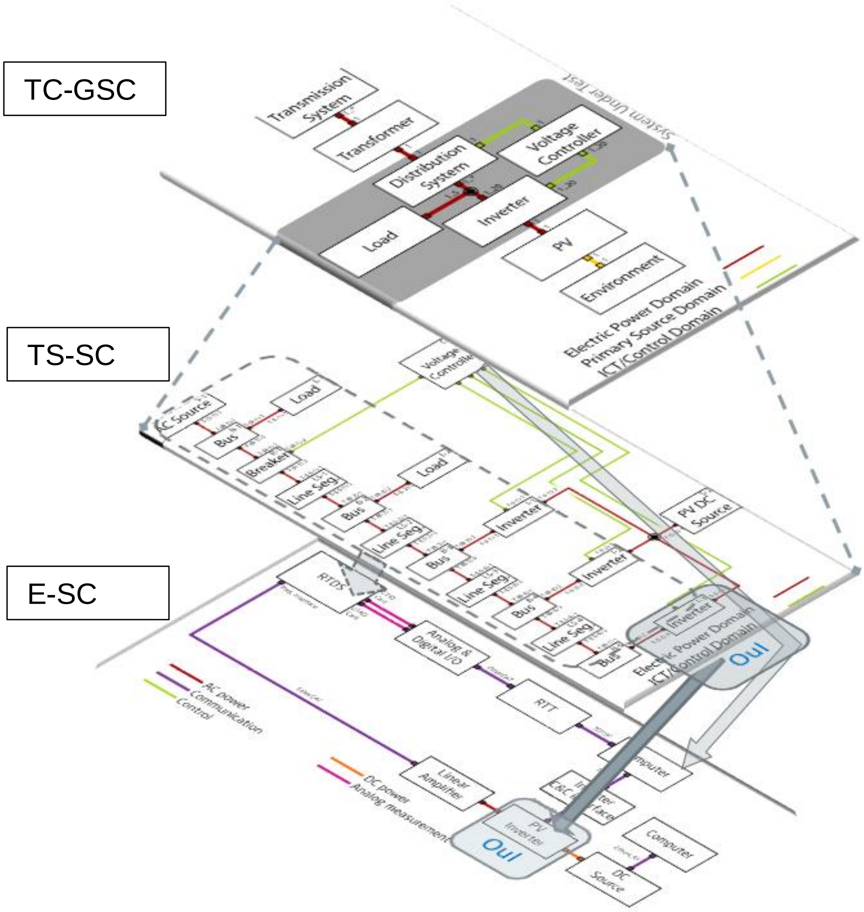


Figure 0.2: Intuitive layering of the TC-GSC, TS-SC and E-SC for the same test description

The test case, test- and experiment specifications include the notions of test criteria and parameters which support the incremental scoping and definition of test factors to facilitate the application of the analytical and statistical methods for experiment design and evaluation. These methods will be used to analytically compose the appropriate test design given the test objective, as well as to enable uncertainty quantification. This will facilitate the exchange and integration of partial test results across RIs and experiments.

ERIGrid's approach on Holistic Testing may be considered as a vision of a pre-standardised process and methodology implementing the testing of a system that includes multi-domain aspects (addressing Power & ICT, P-HIL, as well as heating domains). This vision can be extended to the mutualisation of resources of multiple partners to conduct parallel, sequential and integrated tests according to formalized research infrastructure profiles and mapping procedures.

1 Introduction

Holistic testing relates the idea of fusing together the testing practices from different fields of work, by applying a common and integrated process of testing. This process should allow that systems that exhibit complex multi-domain phenomena can be evaluated and test results have a firm and structured interpretation. The project ERIGrid develops such a holistic testing methodology for the field of Smart Grids.

A core component of such a process is the ability to offer a coherent method for the description of a testing scenario. Many approaches to descriptions of software, systems, scenarios and requirements are available today. In particular, within information and communication technology (ICT), formal specifications of systems and test cases are of widespread use. However, current practice lacks an approach to facilitates specification across ICT and physical domains and that combines these descriptive frameworks in context of a test specification.

In the work of ERIGrid Networking Activity (NA5), a common understanding of test requirements specification has been achieved that can be applied for testing across e.g. electric, thermal and communication technology domains.

1.1 Purpose of the Document

The purpose of this document is to present the ERIGrid approach and method for the description of holistic testing scenarios. The key description concepts, system configurations (SC), use cases (UC) and test cases (TCs) are motivated and described. The sections and appendices of this document are meant to serve independently as background material to inform both ERIGrid internal work as well as Transnational Access (TA) users. The templates provided in this document serve as reference for the remainder of the ERIGrid work.

1.2 Scope of the Document

Taking semi-formal and pre-standard approaches as a starting point, a step-by-step description procedure is outlined that facilitates several research infrastructures to work on one common test case by splitting the test setup into several experiments that can be performed by the dedicated laboratories. The procedure also allows combining software experiments with hardware experiments. Data exchange, test setup refinement, criteria evaluation, and Smart Grid test setup validation receive special attention in this respect. The scope is motivation, definition, introduction and explanation of the basic description concepts. The application of the concepts is only on exemplary level. All specification concepts are explained and the description methodology is outlined. Templates presented in the annex serve as initial guide to the application of the description method. Out of scope is a description of the detailed mapping process required for holistic testing. Also out of scope are the quantification of test criteria and detailed semantics and syntax of domain specific test criteria.

1.3 Structure of the Document

Motivation, core concepts and an overview of the *Holistic Testing Approach* are introduced in Section 2. The concept of multi-domain *System Configuration* is defined and illustrated in Section 3. Section 4 reviews the methodology of *Use Cases* and its application to test requirements. The core of the approach to test specification is presented in Section 5, *Test Cases*. Background, motivation, approach and application examples for the respective methods are presented in each section.

The Appendix presents a *Glossary of Terms*, and a first version of the *Templates* for each specification method.

2 Holistic Testing Approach

In this section we will motivate the need for a holistic testing approach in Smart Grids and Cyber-physical energy systems in general and on a concrete example. Finally, in Section 2.3, the ERIGrid concept of holistic testing and the core terminology are introduced.

The current section outlines the relevant background on testing and defines fundamental concepts and outlines the logical process required for such an approach. The following sections will deepen the relevant methodologies for the formulation of coherent holistic test descriptions.

2.1 Motivation for a Holistic Approach to Testing

An increased utilisation of advanced automation, Information and Communication Technologies (ICT) are transforming the power system to a cyber-physical system. This integration of infrastructures and technology of different domains is driven by several simultaneous developments, within the electricity sector and in the energy infrastructure in general. In the electricity sector, the deployment of renewable energy sources, the liberalisation of the electricity market, and numerous technological innovations significantly impact the structure and operation of the future electric power systems. The application of modern ICT into power systems over the past decade opened up a wide range of development opportunities that, combined, are referred to as Smart Grids. In parallel, increased energy efficiency, removal of fossil fuels and economic drivers are leading to an increased electrification of other energy sectors, such as transportation and heat. In this view the Smart Grid infrastructure becomes part of the wider concept of Smart Cities; here the attribute "smart" then commonly refers to the increased software based networking of all technical components, also termed Internet of Things[1].

Whereas transmission systems are well equipped with sensors and are centrally managed, the integration of heat systems and electric transportation occurs mainly at the level of distribution networks, which used to be operated in a passive way. With smaller units at this grid level, the numbers of systems to be monitored and controlled units greatly increases. While the application of modern scalable ICT systems facilitates this integration, it creates a further coupling of engineering domains that showed little mutual interaction and interdependency before. Challenged by this development, new methodologies and practices must be developed. Viewing the electric energy infrastructure in its entirety as a cyber-physical, critical infrastructure, such new methodologies and practices will have to ensure that the classical high-reliability, real-time operation, and regulatory requirements can be met in the future.

The observed increase of complexity thus manifests in increased coupling across domains, such as electricity, heating and ICT, in scale and heterogeneity. The effects of this increased complexity are not trivial to anticipate, nor to summarize: Operational aspects become a concern, as failures may propagate across increasingly interdependent automation systems, and energy management and coordination can become more challenging. Another challenge with complexity is our ability to conceive, design and develop critical infrastructure systems that depend on such cross-disciplinary competences. Before deployment in an operational environment, Smart Grid solutions have to be validated and tested. Industry and researchers have recognized this challenge and according to the 2015 annual report of Joint Research Center (JRC) of the European Commission, there were 459 projects and demonstrative Smart Grid platforms in Europe with an overall investment of around 3.15 billion €, in the period from 2002 to 2014 [1]. The growing number of Smart Grid research and development projects around the world has led to a significant portfolio of demonstrators and advanced ICT networking features.

[1] To avoiding further reference to vaguely defined terms, in the following, the term Smart Grid will be used to refer to the entirety of this vision of integrated ICT and energy infrastructure.

With this new research and testing infrastructure available, the methodological challenge of conceiving appropriate development and testing principles remains. Historically, the implementation of electricity grid design and extension has been based on primarily electrical considerations, so that considerations that affect for example software design are, if at all, first addressed in a later stage of the design process, when most design parameters are already fixed. While this approach is suitable for conventional power systems, several characteristics of Smart Grids inhibit the applicability of this approach in the future. For instance:

a) *Reliability*: Industrial and conventional communication networks have significant differences in terms of reliability requirements, round trip time, determinism, temporal consistency and hierarchy. For example, for control of physical equipment over industrial networks, the round trip time is expected to be around 250µs - 10ms, while for data processing on ICT network, latency can be extended to 50+ms [2]. It is necessary to take these into consideration, right in the design process, to avoid later reconfiguration.
b) *Cross-domain interactions* that become apparent during the prototyping phase can affect the conceptual design. For instance, assumptions about controls relying on communication between components can yield a system response that requires adjusting the overall design.
c) *Incompatibility due to difference in life spans*: While the lifespan of the electrical assets is generally decades, the governing ICT architecture evolves much faster and may eventually not be compatible with the physical layer anymore.
d) *Cyber-security* issues may crop up at a late stage of the implementation process. Security in the electrical grid is a crucial factor because disruptions in these systems can lead to interruption of critical services and destruction of expensive equipment. Many problems derive from the fact that the classical SCADA systems were not designed to be connected to the outside network infrastructure and security aspects were not considered during the development phase [3]. IEEE standard 1547-2030 [4] identifies and classifies the types of ‘‘intrusions’’ into a substation and discussed the methods for coping with them. Also, guidelines and security measures coupled with electronic controls are discussed in [5],[6]. The Risk Management Framework (RMF) [7] is recommended by NIST (US National Institute of Standards and Technology) as a methodology to implement security control. The IEC 62351 standard for handling security for power system communications defines security profiles for a number of protocols as well as conformance test cases; here, the technical report IEC 62351-12 [8] identifies further resulting resilience issues for cyber-physical DER systems.

These observations motivate a development approach which supports a more integrated, transdisciplinary and potentially iterative testing methodology.

Testing is an integral part of the development process, enabling both incremental quality management and feedback as well as functional verification. The status quo for power systems testing is to focus on a particular Device-under-Test - meanwhile simplifying the behaviour of other components to electrical equivalents. This traditional decoupling raises a question of the global behaviours of the integrated system. A combination of different technologies across domains requires that communication among different specialists is established and founded on the interconnection of different disciplines during the development process. The heterogeneity of Smart Grid domains and technologies, notably the interactions between the various technologies, conflicts with the traditional approach: test laboratories often specialise in a certain domain and can hence only test components for a particular sub-system. So far, testing approaches which combine ICT and electricity domains have had a main focus on individual components [9]. However, in order to support the different stages of the overall development process for smart grid solutions, tests are needed to evaluate the integration on a system level, addressing all relevant test domains [10]. Proposed alternative testing approaches include virtual (simulation) or semi-virtual (hardware in the loop) experiments that cover multiple domains. For these new approaches, questions arise as to whether the test results can be considered valid to draw firm conclusions for a real-world deployment of the tested systems.

A related challenge regards the reproducibility and transferability of test results. Not only the methodology of testing, but also the essential precursors of a reproducible test, the test specifications, are hardly harmonized in the field of Smart Grids. Important steps in this direction have been reported in context of the Smart Grid Interoperability efforts under the European M/490 mandate, in which the methods of system specification have been extended to consider their applicability to the specification of compliance, conformance and interoperability testing [11], advancing in particular the concept of application profiles for interoperability testing. This approach has also been evaluated in experiments of the COTEVOS project[2] for applications to electric vehicle charging. However, such test specifications remain in the ICT domain and do not cover the compatibility of physical device behaviours. There remains a significant gap to bridge to the formal specification of tests that involve both physical device behaviours and ICT functions and components, and that make use of combinations of simulated and physical test platforms.

The collaboration and information exchange among research and industrial institutions has become more and more necessary to efficiently exploit the Research Infrastructures (RIs) and to rapidly transfer new developments. Both for research and commercial testing, stakeholders benefit greatly from access to a shared pool of resources and competences, and eventually, locally or remotely, utilizing infrastructure from several RIs in an integrated fashion. For example, the European project Sophia[3]. With a total cost of 11.5M€, in which only less than 15% is reserved toward transnational access, the project facilitated 42 experiments, hosting more than 1000 transnational access days. Without such collaboration, this research would cost many millions € for infrastructure investment and personnel.

To achieve such gains in context of Smart Grid testing, inter-lab access needs to be combined with cross-lab transferrable procedures applicable to multi-domain and multi-platform testing. A harmonized or even standardized testing procedure is absolutely necessary: only with a shared approach to testing, can stakeholders be enabled to efficiently exploit the capabilities from existing platforms in one RI and to complement the missing capabilities with assets available from RIs at other locations.

Furthermore, connecting interoperable platforms, under a holistic testing framework, will require much less time and resources than locally constructing new necessary experimental modules from scratch. The mutual understanding and control of technological means provide also the possibility to realize multi-site research projects, for example: coupled platforms, long distance energetic management, etc.

The project ERIGrid aims to address the challenges raised above by developing a holistic, cyber-physical, systems oriented approach to testing for Smart Grids. This is being done by creating a platform and methodology for integrating 18 European research centres and institutions. The holistic testing methodology should facilitate conducting tests and experiments representative of integrated Smart Grids by testing and experimentation across distributed RIs, which might not necessarily be functionally interconnected.

Summarizing the challenges outlined above, a holistic testing approach aims to cover the need for:

1) Assessing multi-domain test cases,
2) Bundling and integrating test results from various specialized test laboratories,
3) Clarification of system testing as opposed to component testing,
4) Allowing a combination of virtual and physical experiments, and
5) Development of harmonized and pre-standardised validation procedures.

The following sections introduce basic terminology to describe and illustrate the relevant engineering domains and concepts.

[2] http://cotevos.eu/
[3] http://www.sophia-ri.eu/

2.1.1 Smart Grid Intelligence at Different Levels

Distributed intelligence is one of the key drivers requiring advanced testing, as the system behaviour is more strongly dependent on the interactions of remote components. The distribution of intelligence in the Smart Grid can be illustrated on different levels as shown in Figure 2.1.

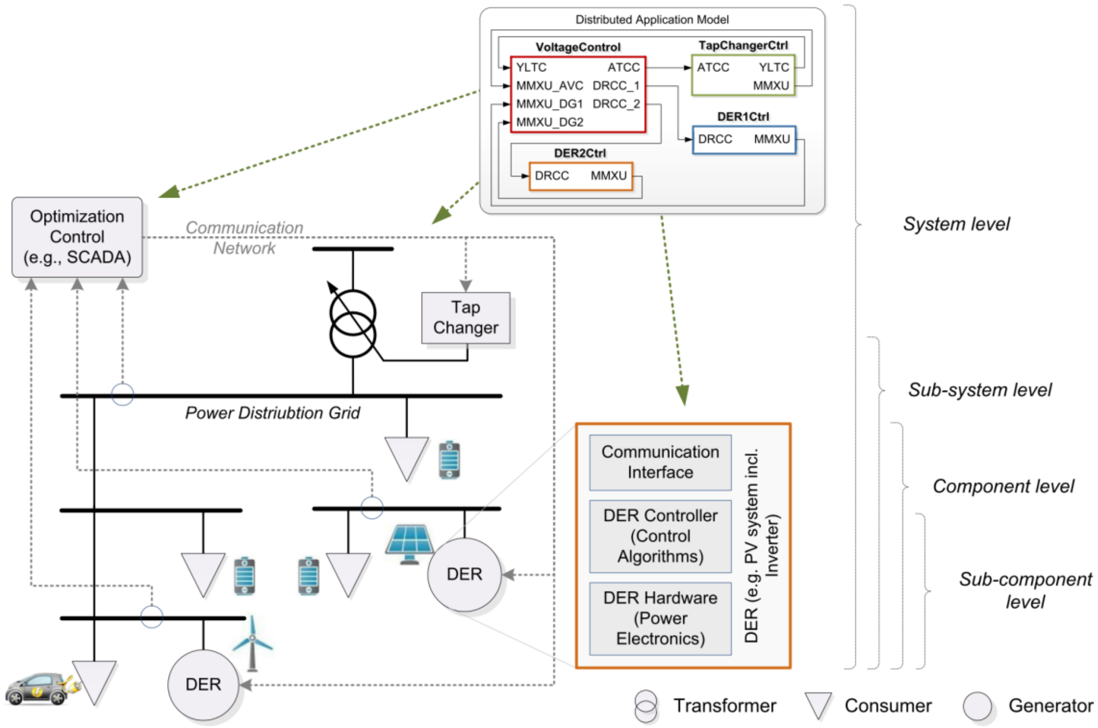


Figure 2.1: Intelligence on different levels applied to smart grid systems (adopted from [12])

A brief categorization of these different levels has been proposed in [10], as follows:

- *System level:* Operational approaches like power utility automation, demand-side management or energy management are tackled by this level. Functions and services of the underlying sub-systems and components are triggered in a coordinated manner for execution from a systems perspective. Both, central as well as distributed control approaches are used on this level.
- *Sub-system level:* The optimization and the control of sub-systems are carried out below the system level whereas the corresponding functions, services, and algorithms have to deal with a limited amount of components (DER, energy storage system, electric vehicle supply equipment, etc.). Examples for this level are micro-grid control approaches and home/building energy management concepts. Also an energy storage system together with a distributed generator installed at the customer side can be considered as a sub-system. Distributed automation and control are typically applied.
- *Component level:* Distributed Energy Resources (DER)/RES, distributed energy storage systems but also electric vehicle supply equipment is covered by this layer. Components typically provide advanced functions like ancillary services. Intelligence on this level is either used for local optimization purposes (device/component behaviour) or for the optimization of systems/sub-systems on higher levels in a coordinated manner.
- *Sub-component level:* Intelligence on this level is mainly used to improve the local component behaviour/properties (harmonics, flicker, etc.). Power electronics (and their advanced control algorithms) is the main driver for local intelligence on this level. The controllers of DER, distributed energy storage systems, electric vehicle supply equipment and other power system components (tap-changing transformers, FACTS, etc.) can be considered as examples for sub-components.

Major requirements for the realization of ICT/automation systems and component controllers are flexibility, adaptability, scalability, and autonomy. Furthermore, interoperability and open interfaces are also necessary to enable the above described functions on the different levels [12].

2.1.2 Design and Development Process of Smart Grid Solutions

The design and development process of Smart Grid solutions covers several stages, and each stage involves testing with different requirements to the testing methods. The stages are mainly dependent on the applied system engineering approach or process model (V-model, etc.), but also on the overall complexity of the system under development. In general, the following four main design stages can be observed during the whole development process [10]:

- *System-level requirements and basic design:* During the first design stage usually the system-level requirements and application scenarios are being identified. In the following a basic design and high-level architecture specification are typically been carried out.
- *Detailed design:* After the conceptual design has been elaborated a detailed design and engineering of the system under development is done. Functions and services are also identified and specified.
- *Implementation and prototype development:* During this development phase first prototypes are being developed. The process of transforming a concept into a prototype often introduces issues which were not considered during the design stage(s). Often boundary problems like communication latencies or nonlinearities are neglected during the first versions of a basic concept. During the development of a prototype iterative refinements of solutions/algorithm are often necessary.
- *Deployment and roll out:* This stage mainly covers the realization of a product as well as the installation/roll out of components and solutions in the field.

Compared to other domains, challenges during the design and development of smart grid solutions are *(i)* the fulfilment of high-reliability requirements, *(ii)* the observance of (strict) real-time requirements, *(iii)* the compliance with national rules, and *(iv)* the interaction with several system integrators/manufacturers.

2.1.3 Motivational Example of a Test Scenario

An example of a test scenario that could be of interest for holistic testing is illustrated in Figure 2.2, which outlines a possible laboratory test setup. The test setup includes lab components such as a digital real-time simulator with analogue and digital I/O, a linear power amplifier, and PV simulator (a DC source), as well as several communication and computation units, and a commercial PV inverter. The smart grid solutions being evaluated in this setup is illustrated in Figure 2.1: an On-Load Tap Changing (OLTC) transformer is used in this Smart Grid solution together with reactive and active power control provided by DERs and electric storages. The goal of this application is to keep the voltage in the power distribution grid in defined boundaries due to distributed generation and to increase its hosting capacity with a high share of renewables [13]. The corresponding control approach has to calculate the optimal position of the OLTC and to derive set-points for reactive and active power which is communicated over a communication network to the DER devices and electric storages [10].

The setup reflects both ICT and electric power components, and illustrates the linkage and interdependency of these components, as it combines controller hardware in the loop (CHIL) with power hardware in the loop (PHIL) and remote communication. For the test scenario, there is further a need to define which use case are covered by the test and what the goal of this test should be. As core use case, the “Optimal centralized coordinated voltage control” is suggested (reported in Annex 9.4 “Use Case Definition Example”), which defines a control function placed in a Central Controller, receiving grid state information as inputs and setting active and reactive power setpoints for remote tap changers and PV inverters.

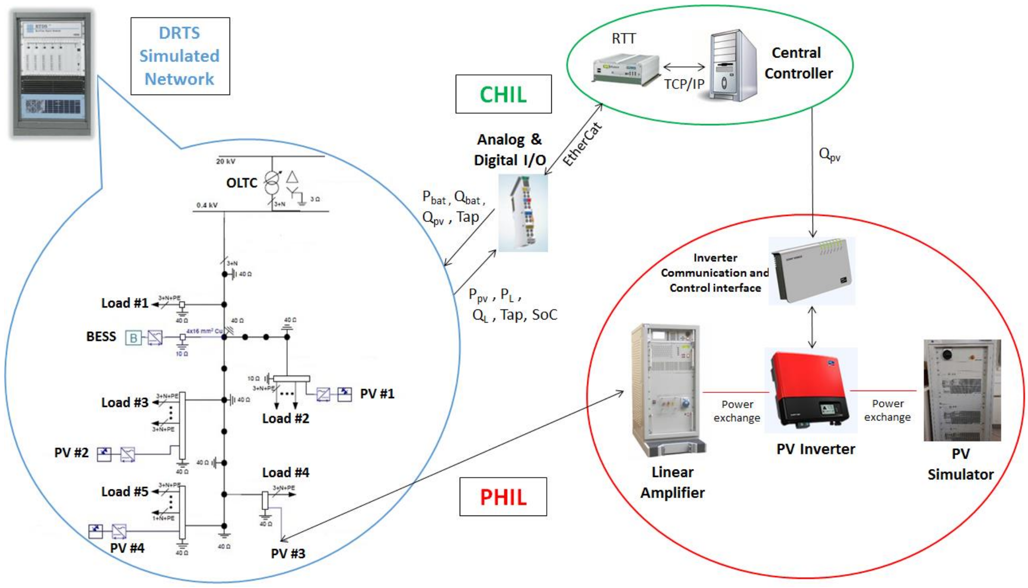


Figure 2.2: Intuitive graphical representation of a holistic test scenario with CHIL and PHIL

For an expert in the field, the test scenario is intuitively clear: the real-time simulator emulates the grid behaviour so that the closed-loop effect of the central controller can be evaluated, and a physical PV inverter is included to evaluate the effect of non-ideal component behaviour. Then again, the test setup also raises a question: what is being evaluated, the PV inverter, or the Central Controller? Correspondingly, it is not clear which component is the object under test, and which aspects of the setup is meant to *emulate* a real-world situation, and. And further the test setup leaves assumptions and test goals unstated, such as: is the communication between controller and inverter assumed to be ideal? What are the evaluation metrics? Which real-world phenomena are covered by the test scenarios and models?

Such and other questions need to be resolved by a test scenario description. After reviewing the state of the art in brief, we turn to outlining the proposed ERIGrid approach to test description in Section 2.3.

2.2 State of the Art of Testing Approaches and Methodologies

Providing a relevant context for the ERIGrid proposed approach, this section offers an overview of related methodologies in several engineering domains. First, the role of testing in engineering processes is reviewed, identifying common test types and purposes in engineering processes. Next, testing technologies and methodologies that are common in a power systems and automation context are summarized, including those that entail a virtualization and simulation of test components in Section 2.2.1. Finally, Section 2.2.2 discusses related theory on systematic test evaluation.

2.2.1 Testing and its Role in Engineering Development

To anticipate an integration of testing methodologies practiced in different engineering domains, the use and purpose of testing must be understood in context. First the role of testing and test specification are reviewed in context of development processes, second, the testing role in different engineering domains are summarised.

2.2.1.1 Development processes and the role of testing

A classical approach to development is to adopt a sequential engineering approach where the expertise of specialists is compartmentalized. For more complex modern systems, interactions across different technologies can influence the overall design significantly, so that a sequential approach of stepping through engineering disciplines becomes slow, even inadequate: the different actors in design must communicate with each other to avoid interfacing problems between different parts and aspects of the system. A number of structured approaches to system development have been proposed and employed in practice. Testing is a critical component in all of these – however, it is the interaction between design, development and testing that characterizes the differences between approaches.

For software development, the core sequence has been noted as the Systems Development Life Cycle (SDLC), which outlines a recurring sequence of requirements analysis, design, implementation, integration and testing, system deployment, operation; the steps of deployment and operation are at times referred to as "evolution", emphasizing the probable iterations of a design. It has been noted early that a pure 'waterfall' sequence is highly unlikely to lead to a satisfactory product, or even to deliver at all [14].

Testing has a central role in all engineering processes in that it forms the gateway to the conclusion of any development step. There are therefore many types of tests, each corresponding to a different type of development effort [15]. A common structured form of characterizing these stages is formulated in the V-model of system development: the left arm of the V represents the specification and development steps; the right arm represents the corresponding testing steps: in a sequence of tests from simple to complex: *Unit testing* evaluates immediate correctness of code units, *integration testing* evaluates the compatibility of code modules, *system testing* evaluates against the design and architecture specifications, and *acceptance testing* evaluates the complete product with users, thereby systematically evaluating all initial requirements. For each of these test to be carried out, a test design must be specified. The V-model therefore provides a key reference to understanding the mutual dependence between specifications and testing[4]. Yet the V-model still represents a sequential approach to testing, which is inadequate to highly complex systems.

In the spirit, of today's multi-disciplinary work and the faster development cycles of modern systems development emphasize concurrent engineering work. In *concurrent engineering [15]*, different tasks are tackled at the same time, and not necessarily in the usual order. Ideally, this means that knowledge that usually was found out later in a process can be discovered in earlier stages of the development, improving the product design while also saving time. The design procedure in concurrent engineering generally adopts the V-model, casting it into more circular processes motivated by the SDLC. An example is the W-model, where testing activities are coupled directly to the corresponding design activities. Rather than to focus on specific dynamic test stages, as the V-Model does, the W-Model focuses on the development products themselves. As a consequence, the W-Model of testing focuses specifically on the product risks of concern at the point where testing can be most effective.

Testing plays an important role in concurrent engineering. As single-domain systems evolve into *multi-domain* and sequential development turns into *concurrent* engineering, the testing technologies and methods must also evolve. While the testing process was previously focused on verification of one aspect of the system, it is nowadays required to consider the interaction among the different domains of a multidisciplinary systems and the final design system as a whole. This new aspect of testing arises because it is not trivial to deduce the global behaviours of the system from the properties of its constituent parts. A combination of different technologies is optimal if and only if a real communication among different specialists is established and the interconnection of differ-

[4] A related interpretation of the V-model emphasizes the "validation and verification" aspect in the testing. The difference between these terms will be discussed later in this report.

- Waterfall: requirements are clear from the start, which allows a sequential development
- V-model (validation and verification model): integrates testing more tightly into the development
- W-model (or dual V model): every module in the V-model is tested and validated
- agile: iterative approach commonly used when the requirements are unclear from the customer side, and the system is developed iteratively [16]

ent disciplines is taken into account in the design process. The heterogeneity of Smart Grids, notably the interactions between the various domains, requires a holistic testing procedure.

Intuitively, the concept of a holistic testing procedure should support design methodologies used in *concurrent engineering,* thus facilitate a wide range of tests at different levels of component and systems maturity. What remains unclear here is how this approach facilitates the engineering of *multi-domain systems*.

Test Types by Test Object

In summary, depending on the object of the test, we can distinguish the following three levels:

- *Component testing*: A unit is the smallest testable part of an application. In our scope a unit could be considered as a single device or component. Component testing focuses on each component individually. Component test can be distinguished into white-box testing and black-box testing. *White-box* testing is typically applied in earlier engineering stages on prototypes, whereas *black-box* testing is performed at a later stage, applied to a completed product or component.
- *Integration testing*: In integration testing separate units (systems, devices) will be tested together to expose faults in the interfaces and in the interaction between integrated components. Integration testing can validate the system interoperability for the specific systems being integrated, but does not provide reference guidance as to interoperability with other systems of a similar type. Integration testing derives interoperability testing.
- *System testing*: System testing is conducted on a complete, integrated system to check if the integrated product meets the specified requirements. It looks at the system from the perspective of the customer and the end-user. The system test requires no knowledge of the inner design of the code or logic. The conformance of the device according to the specified grid requirements must be able to be proven by an independent party. System testing can be used to validate a model for the electrical behaviour of the device. The model can be used for a further simulation in the project level.

Test Types by Test Purpose

With respect to the evaluation of a set of requirements, such as requirements from a standard, the following test types are distinguished:

- *Prototype testing* (white box testing). The purpose is to verify the internal operations by testing every physical process (or virtual path in the case of software) within the particular component.
- *Conformance Testing*: Determines whether an implementation conforms to the full profile (standards, norms) as written, usually by exercising the implementation with a test tool.
- *Interoperability Testing*: Connects two or more implementations together and determines whether they can successfully interoperate.

Interoperability is significantly different from conformance because it is often possible for two systems that comply to a standard to be unable to interoperate. These situations can arise because they have chosen different or conflicting options within the standard or because the implementations have conflicting interpretations of the specification.

Standards aimed at interoperability are typically framed for communication purposes. An even higher level of requirements is associated with *interchangeability* of devices, as here also the physical dynamics have to be evaluated [17]. Further distinctions can be introduced with respect to the stages within a development process.

2.2.1.2 Integrated testing and development processes in Mechatronics and ICT

ERIGrid's view on testing extends to multi-domain and parallel testing (combining results from independent tests), and the systems to be tested often are composed of a number of subsystems or components that could be tested individually and in cooperation. Multi-domain testing is common in mechatronics, which includes electronics and mechanical systems, e.g. utilized in the automotive domain. The development and testing of more complex *systems* and *systems-of-systems* is common in the ICT (both IT and OT) domains.

Information and Communication technology

A comprehensive review of IT domain testing models can be found in [18]. Some models have recently been readapted into modern development models, such as Test-driven development [19]. In the IT domain, there are "Holistic testing procedures" which can tentatively be classified into two types:

I. *A complete feature test of software (regular check) [20]:* The testing procedures of the first type aim at checking the compatibility and compliance of a software/application to different environments and usages. The validation checks of normalization organizations are examples of this type (e.g. the w3c RDF validators [21]). However, the procedure varies from test to test. These tests have the purpose of enabling the quality management during development. No standard has been recorded[5].
II. *A test including crowd testing and expert testing [22] (normally before beta phase):* The second type of holistic test procedure is a mixture of crowd testing at the front end and expert testing at the back end.

The *Holistic testing* in the IT domain does not include the *multi-domain* aspect but rather a complete verification of a software (features and compliance to executing environment).

Another important integration in Smart Grids is associated with standardization and harmonization of interfaces between different systems. The concept of Interoperability, and the corresponding Interoperability testing is testing aimed at system integration. An alignment of interoperability testing with standards specification via Interoperability Profiles (IOP) has been proposed by the Smart Grid Coordination Group in [17].

Testing and development approaches in the IT domain are further examined in Section 5.1.4, where concepts within software testing, e.g. test-driven software development, including the communication interoperability testing standard TTCN-3.

Mechatronics

In the mechatronic system design process, possible technical solutions are determined in order to satisfy a set of requirements. A mechatronic system is a juxtaposition of different physical domains, typically controlled with embedded systems. Thus, the classical methods of mechatronic system design traditionally used a sequential approach to development. In this view the V-development sequence is commonly used. The decoupling of different technologies in this design strategy however poses questions over the association of the sub-systems, as the different domains in a mechatronic system interact in synergy. As mentioned previously, *Concurrent engineering* is the modern approach also applied in mechatronics that is based on the communication the different project partners and the implementation of common tools and platforms of development [23]–[25].

A number of design methodologies have been proposed in literature. Roughly speaking, they are classified into two categories: direct methods or forward methods (trial -> error -> correction) and

[5] TTCN-3 is a standard test description language. Its purpose is to describe conformance and interoperability tests but does not define any testing procedure.

inverse methods (inputs are deduced from desired outputs). Direct methods are very popular and are applied at large scale in industry. However, they are expensive in computational time. The inverse method provides designers with a solution requiring less computational effort for system sizing problems. However, it may be difficult to apply, since an invertible model is required.

Applications such as robust design, uncertainty and sensitivity analysis, robustness or even reliability analysis may employ the Design of Experiment[6] (DoE) methodology (which can be seen as a kind of direct approach). The direct or forward approach is often used in such problems as uncertainty analysis [26], robustness analysis [27], μ analysis [28], and assembly tolerance synthesis [29] or sensibility analysis [30]. The inverse approach for system design, on the other hand, may involve virtual models, which may serve solely for calculation purposes and may not follow causal rules. Inverse problems are popular in such applications as sizing [31], tolerance synthesis [32] and command synthesis [33]. The testing is always in the form of a direct approach, because the final designed system must follow physical causal rules.

2.2.1.3 Integration of ICT and Automation components in power system testing: virtual prototypes and hardware-in-the-loop

The field of power systems acknowledges the shortcoming of the classic component testing (further discussed in Section 5.1.3) and is moving towards system tests that integrate ICT. Along with the rapid revolution in the ICT domain, the field of power system testing has also evolved in order to adapt to new technological developments. Figure 2.3 shows this evolution and the concept of power system testing.

The concept of model-based design appeared to facilitate and accelerate the process of product development. It provides the gateway to a framework of rapid prototyping, testing and verification. Not only is the testing and verification process enhanced, but also, hardware-in-the-loop simulation can be used with the new design paradigm to perform testing of dynamic effects on the system more quickly and much more efficiently than with traditional design strategies.

The core element of the model-based design paradigm is the replacement of real tests via prototype by offline simulation and real-time simulation. Simulation allows specification, requirements, and modelling errors to be found at the design phase, rather than later in the development effort. Real-time simulation can be done by automatically generating the simulation code for the developed equipment from a computer model. This code can be deployed to a special real-time prototyping computer that can run the code and emulate the operation of the design object. Similar to the HIL approaches outlined above, also in mechatronics engineering, real-time simulation and virtual prototyping are very common, often earlier in the design process.

While modelling of complex systems may appear to be a challenge and subject to errors, the engineer may use a Hardware-in-the-loop (HIL) approach. A HIL, or more specifically, Controller-Hardware-in-the-loop (C-HIL) simulation must include electrical emulation of sensors and actuators. These electrical emulations act as the interface between the plant simulation and the embedded control equipment under test. The values of emulated sensors are controlled by the plant simulation and read by the embedded equipment (feedback). Likewise, the embedded equipment implements its actuation signals by outputting actuator control signals on electrically emulated actuator interfaces of the real-time simulator. Changes in the actuator control signals then result in changes in the plant simulation, thus closing the loop. This test scenario therefore provides a realistic closed loop test result – to the extent that the simulated plant accurately represents the real plant and its sensors and actuators. This approach is suitable in particular when a prototype is not available, or testing on the prototype is dangerous or expensive.

[6] Design of Experiments is further discussed in Section 2.2.3.

To decide whether a HIL approach or testing on the physical plant is more appropriate, the test efficacy should be evaluated. Metrics related to development and testing efficacy are typically formulated using factors such as: Cost, Duration, Safety and Feasibility.

The cost of the approach should be a measure of the cost of all tools and effort. The duration of development and testing affects the time-to-market for a planned product. Safety factor and development duration are typically equated to a cost measure. Specific conditions that warrant the use of HIL simulation include the following:

- Enhancing the quality of testing (i.e. more test conditions can be simulated)
- Tight development schedules (i.e. faster design iterations)
- High-burden-rate plant (i.e. a very expensive unit to take time for testing)
- Early process human factor development (e.g. by offering a ‘realistic’ basis for experimenting with operator support systems)

The new technologies of virtual prototypes and hardware-in-the-loop enhance the capacities of testing of a complex system, while reducing time and cost. They also provide the means to test a system as a whole, including the interaction and communication among different domains. This integrated testing cannot be achieved by conventional testing technology.

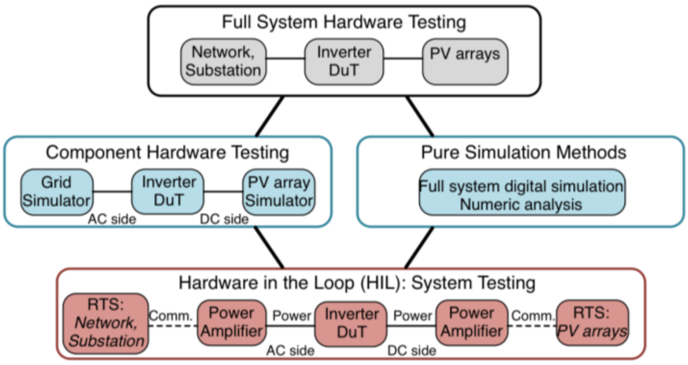


Figure 2.3: Power System testing with P-HIL

2.2.2 Systematic Test Design and Evaluation Methods

The concepts of hardware-in-the-loop tests and virtual tests are discussed in previous sections. In order to carry out these kinds of tests, accurate mathematical models for components and systems must be derived, such that they can simulate the relevant parts of the system for a given test. Given that one of the previously mentioned motivations for the ERIGrid project is to allow for combinations of virtual and non-virtual tests across research infrastructures, it is relevant to investigate how the required system and component models can be derived.

In [34] an experiment is defined as a “series of tests in which purposeful changes are made to the input variables of a process or system so that we may observe and identify the reasons for changes that may be observed on the output response”. Within the engineering method, experiments are often used to create models of the performance of the system, i.e. empirical models. These models are useful when the complexity of the system is such that a mechanistic (white-box) model is impossible to derive. Design of experiments (DoE) is a systematic method to determine the relationship between factors affecting a process and the output of that process. In other words, it is used to find and quantify cause-and-effect relationships using statistical methods. This information is needed to generate empirical models.

In the vocabulary of DoE, inputs that affect the outcome of an experiment are called factors. Since most systems have more than one factor, experiments must be designed such that both the impact of each factor and the impact of varying factors together are analysed. Experiments designed with this in mind are called factorial experiments and are characterized by aiming at making the most efficient use of the experimental data. In factorial experiments factors are varied together over levels or ranges.

Three concepts lie at the heart of the DoE methodology: *randomization*, *replication* and *blocking*. Randomization means that both the allocation of the experimental material and the order in which the test runs are executed must be random (in order to avoid biases). Replication means an independent repeat of each factor combination test (note that this is different from a repeated measurement). Blocking is a technique used to improve the precision with which comparisons among the factors of interest are made.

A general guideline for DoE is presented in [34] consisting of the following 7 steps:

1. Recognition of and statement of the problem
2. Selection of the response variable
3. Choice of factors, levels and ranges
4. Choice of experimental design
5. Performing the experiment
6. Statistical analysis of the data
7. Conclusions and recommendations.

An aspect of step 6 which is especially relevant is the estimation of the uncertainty propagation from the inputs to the outputs of the system. With a correctly designed experiment, we should be able to estimate reliability and validity of our results, and have a characterization of how variance in our inputs (factors) affect the output (response). Also, in this step the formulation of empirical models based upon the experimental results is done.

Frequent applications of DoE are:

- Identification of relevant (significant) parameters/factors
- Identification of “optimum” parameter values
- Identification of robust parameter bands
- Constructing a (meta)model of the functional input-output- relation(s)

It is clear from this short description, that the DoE methods have a high relevance to the composition and evaluation of a holistic system test. The concrete potential for application of this method toward holistic testing of Smart Grid systems requires further assessment. A follow-up on this description is provided in Section 5.1.2.

2.2.3 Relevance and Conclusion

In principle all types of testing are relevant to ERIGrid, but not all types push the requirements of the holistic approach. The formalization of ICT based approaches is very relevant to a holistic approach, as common semantics and ontologies facilitate a shared understanding of test goals and related assumptions and resources.

In principle, a holistic testing method should support any of the established development strategies and testing purposes discussed above. However, as for example Interoperability Testing has been subject to standards development (e.g. [17]), where emphasis is placed on the formulation of test requirements on the basis of Interoperability profiles (IOP). An outcome of this work is a systematic strategy for the derivation of generic IOP from standards as “Basic Application IOP” (BAIOP). In contrast to this development ERIGrid places emphasis on the formulation of the concrete lab test specifications, which a) go beyond interoperability testing, and b) assume a given set of test requirements – based on standards or not.

It can be expected that for the ERIGrid participants, the test focus will be on tests that involve both physical dynamics and ICT component behaviour, extending the testing needs both to early development stages and to compliance testing. The assessment of full standard conformance is, in contrast, out of scope.

A summary of the above discussed testing methods and overview in relation to suggested usage across the design and development process (as described in Section 2.1.2) has been provided in [10] and is reported in Table 2.1.

Table 2.1: Brief overview of validation approaches used in power system engineering [10]

	Requirements / Basic Design	***Detailed Design***	***Implementation / prototype***	***Deployment / Roll Out***
Simulation Methods	+	++	o	-
Lab Testing Methods	-	-	++	+
HIL Approaches	-	-	++	++
Field Tests and Pilots	-	-	-	++

Legend: - … less suitable, o … suitable with limitations, + … suitable, ++ … best choice

Testing procedures and experience from the fields of power systems, mechatronics, and ICT domains will be employed as a practical foundation for the ERIGrid work. The integration of simulated (virtual) and physical component testing, along with the execution of test-software on special purpose embedded systems as described in Section 2.3.3 challenge our notions of test specification. A coherent and systematic description and distinction of the full test system and the actual test object is challenging: the virtualization of test components into simulations or other embedded realizations shifts the scope between the “test system” (as the hardware or software being tested and emulated in a test) and the enabling “test setup” (as the lab hardware or software providing the execution environment of a test). There is a lack of description tools to facilitate the delineation of these essentially opposing roles in testing, which must be addressed.

To formulate a holistic testing approach, it will be important to arrive at a test specification that can be compliant with the notions established in the DoE method, so that meaningful factors can be derived.

2.3 The ERIGrid Holistic Testing Approach

A holistic Smart Grid research and development approach not only has to address the whole development cycle (design, analysis, simulation, experimentation, testing and deployment) but also has to take into account all relevant components, facets, influences that future power systems will comprise of, which might interact with the controller, algorithm(s), or use case in question. Testing highly integrated systems is invalid without taking into account possible disturbances by users, markets, ICT availability, etc. Formal analysis of these vastly complex, integrated systems is not yet – if at all – possible.

Hence, rigorous testing strategies are required that allow for the validation of integrated systems of different domains represented at different Research Infrastructures (RIs). Due to the importance of the system at hand and immaturity of controllers, applications, or hardware, real-world embedded field tests are in many cases out of question.

Although a functional integration of the aforementioned RI running in parallel and yielding integrated holistic energy systems is theoretically possible it remains practically infeasible for the full spectrum of required test. In order to be capable of conducting tests and experiments representative of integrated Smart Grid systems, testing and experimentation must be possible across distributed and not necessarily functionally interconnected RI. The outcomes from experiments at different RI are dependent on each other and must be analysed in an integrated way.

ERIGrid proposes an approach to realize a holistic procedure for Smart Grid system validation to support comparability between experiments of different setup and design, facilitate subsequent re-utilization of experimental results from different stakeholders through consecutive, sequential and parallel experiments. The goal is thus to provide a theoretical and practical framework for:

- Assessing multi-domain test cases,
- Combining the expertise of the various laboratories,
- Integrating of virtual prototype and hardware-in-the-loop experiments for Smart Grid validation and roll-out.

In the following sections we outline the fundamental concepts of the proposed test scenario description and specification procedure.

2.3.1 Definition of Holistic Testing

While the proposed methodology is based in existing standards, the concept of holistic testing remains vague. We therefore define tentatively:

> **Holistic testing:** *The process and methodology for evaluation of a concrete function, system or component within its relevant operational context with reference to a given test objective.*

Here, the term "process" refers to a goal oriented sequence of steps, and the corresponding methodology to how these steps are to be carried out. For the document at hand the focus is rather to provide the means of specification to support this holistic notion of testing and experiments. In this view, holistic testing requires a multi-domain approach to encompass the full operational context of typical Smart Grid functions, including for example a combination of electrical- and ICT sub-systems.

While some aspects are common to both, for testing in a pure IT context, and in physical laboratory tests, the procedural and description requirements differ widely. The concrete steps and notions employed in each domain are not necessarily transferrable. Another aspect of the holistic testing approach is the merger of different cultures of testing, which can be portrayed as:

- a device-oriented culture of physical testing and
- a culture of testing ICT objects such as implementations of protocols and algorithms.

Rigorous formal specification of test cases as well as automated execution of tests are common in the ICT domain [35]. In the testing of physical components, the test object is delimited by its physical boundaries, requiring little further formalization of the test object. Rather than formal knowledge, the interpretation of physical test specifications requires domain specific insight and physical understanding, often implicitly known by the test engineers. Test specifications therefore tend to be domain specific, less formal, while requiring a significant amount of contextual knowledge. Further, much of a test design is decided by the available test setup.

A challenge is therefore to formalize the complete cyber-physical system context and test criteria, in a common framework. A balance must be struck between the development of theoretical underpinnings and the suitable complexity for a practical framework:

- too little formalization is likely to cause unclear specifications and ambiguous results,
- too much formalization, on the other hand, will render the framework impractical.

Our goal is therefore *first* to identify key principles, formulate strict theoretical underpinnings and definitions: this will allow for clear distinction criteria where specifications are in question and require refinement. These underpinnings then are also meant to provide the foundation for a future formalization of the overall holistic testing process, where appropriate. *Secondly*, we aim to provide practical guidelines and examples of the specification process, avoiding overly formal descriptions.

Observing the above definition, we can begin by offering the following refinements:

- In contrast with a more conventional notion of “equipment under test”, or an “Object under Test”, our definition identifies two critical system boundaries relevant for a test specification:
  - the *Object under Investigation (OuI)*, corresponding to the classical “object under test”, this term refers to the component, system, or function that is being investigated
  - the *System under Test (SuT)*, encompasses the wider “Operational Context” of the OuI, the surrounding systems and subsystems required to emulate the relevant direct and indirect interactions and constraints a OuI undergoes in the scope of a test.
- There needs to be an explicit selection of the test object to take the form of a *function*, *system* or *component* - each to be treated by distinct approaches to representation and test infrastructure integration.
- The notion that interactions and constraints occur within and across domains (such as electric power and ICT) requires a representation of *system configurations* suitable for multi-domain experiments.
- The test objective is an important factor in motivating the test, but also framing the object under investigation; to demonstrate the direct linkage to the object under investigation, we call the refined test objective *purpose of investigation.*

Based on these refined concepts we can reformulate the definition above to the following detailed version:

> **Holistic testing** (detailed)**:** *The process and methodology for the evaluation of a concrete function, system or component as object under investigation within its relevant operational context given by the system under test, corresponding to a purpose of investigation.*

In addition to the above introduced refined terms, several the main concepts employed are rooted in well-established engineering terminology:

- **Use case:** Class specification of a sequence of actions, including variants, that a system (or other entity) can perform, interacting with actors of the system [36].
  - *Remark:* Use cases motivate *functions*, goals, and performance criteria relevant in particular to ICT and control aspects of a test;
- **Component:** the constituent part of a system which cannot be divided into smaller parts without losing its particular function *for the purpose of investigation* (adapted from [37]).
  - *Remark*: In a system configuration, components cannot further be divided; connections are established between components.
- **System (generic):** a set of interrelated elements considered in a defined context as a whole and separated from their environment [38, p. 600].
  - *Remark*: In a system configuration, a system represents a grouping of components, which may be divided into sub-systems; interfaces between systems are called connections.
- **Domain:** An area of knowledge or activity in the context of Smart Grids characterized by a set of concepts and terminology understood by practitioners in that area [36].
  - *Remark*: In a system configuration, domains represent a categorization of the connections between systems; a domain can be divided into sub-domains; domains interface with other domains via components.
- **System(s) configuration:** an assembly of (sub-)systems, components, connections, domains, and attributes.

While these concepts are familiar, in particular their application in a framework of holistic testing requires further definitions and illustration. Therefore, Section 3 is dedicated to the proposed approach to the representation of multi-domain System Configurations for the formulation of both conceptual test cases and concrete experiments. Correspondingly, Section 4 is dedicated to the Smart Grid application background and ERIGrid approach to the representation of Use Cases.

Finally, the ERIGrids main ambition of this report is to provide a careful account of the required specifications in a holistic testing process. Whereas the details of this approach are provided in Section 5, a few definitions here will help orientation through the respective report sections.

We therefore introduce *three levels of test definition,* where each references the previous level, leading to an incremental scoping of a concrete test/experiment[7].

1. A **test case** provides a *set of conditions* under which a test can determine whether or how well a system, component or one of its aspects is working given its expected function.
2. A **test specification** defines the test system (i.e. how the object under investigation is to be embedded in a *specific system under test*), which parameters of the system will be varied and observed for the evaluation of the test objective, and in what manner the test is to be carried out (test design).
3. The **experiment specification** defines by what exact means a given test specification is to be realized in a given laboratory infrastructure.

In analogy with a Use Case, a *test case* formulates key objectives and context of a test, whereas the further steps of specification provide a concrete foundation for the eventual test execution. The test case defines by the test objectives, which are derived from the context provided by the development process of the test object:

Test objective: The purpose for carrying out the test. These can be divided into three categories:

- <u>Characterization test:</u> a measure is given without specific requirements for passing the test. *Examples*: characterizing performance of a system; developing a simulation model.
- <u>Validation test:</u> functional requirements and abstract measures are provided, but are subject to interpretation; qualitative test criteria.
  Example: is a controller ready for deployment?
- <u>Verification test:</u> Tests where requirements are formulated as quantitative measures and thresholds of acceptable values are quantified.
  Example: A test evaluating whether a component conforms with a given standard.

With the conditions for a successful test defined, the next step is to identify what concrete object is to be tested, and how such a test is to be carried out:

A *test specification* aims to clarify the relation between an object under investigation, test objective, and the configuration, means and method under which a test is to be carried out and evaluated (i.e. test system and test design). Related to the test specification is a system configuration that defines the details of the system under test including the object under investigation, as well as the simplified interfaced elements at the SuT boundaries, offering a concrete quantitative formulation of the test objective.

Test System: The specific system configuration of a System under Test that conforms with the (generic) identification of the System under Test of a related test case, implements all Functions under Test, reflects all identified Domains of Investigation and includes all relevant Objects under Investigation.

[7] It should be noted that the terminology can be misleading: The interpretation of “test” vs. “experiment” is that the latter is more concrete and the former abstract. This distinction originates from the ERIGrid DoA, and is therefore rather historic. It can be associated with the idea that the “Design of Experiments” methodology may be only applied to the concrete experiment. This, however, is not the case: the test specification already selects input and output parameters. Similarly, “test setup” is taken to mean approximately the same as “experiment setup”; however, in this report, “test setup” is used to refer to a common sense notion of a test setup, and “experiment setup” refers to the here defined notion of a RI specific system configuration for a specific test.

Finally, the *experiment specification* identifies the concrete laboratory process, components and devices required for executing a test, in which configuration the test system is represented by the available lab components and systems. The experiment specification defines the actual test setup, here called *experiment setup*, as a system configuration:

Experiment setup is a (cyber-)physical system configuration that represents all the test-relevant aspects of the complete SuT including the *actual* OuI realised within a test environment.

Whereas an experiment setup can be described using the same description methodology as used for identifying a SuT, the concepts are entirely distinct. Apart from the OuI, any component of an experiment setup is only a representation of one or several SuT components. We speak of a "mapping relation" between SuT and experiment setup. The test environment can be a single RI, but in principle also a combination of RIs if these are interconnected as part of one experiment execution.

2.3.2 ERIGrid Overall Holistic Test Description and Evaluation Procedure

The concepts outlined above are central to the holistic test description method. The complexity of the concepts may not seem justified for a single component test; however, they are required to offer the description flexibility required for different variants of testing purposes and test realizations. To facilitate their application, a reference procedure is outlined here, which illustrates the relation and application of the concepts in test description practice, and in reference to the remainder of this document.

The main steps of the ERIGrid approach to holistic testing are outlined in Figure 2.4. It can be seen that the basic steps correspond to an *incremental specification* of what the actual test subject will be, as represented by steps 1 – 4. This incremental specification has been identified as necessary to separate the available research infrastructure from the identification of a test case.

The main system functions and the object under investigation should be identified early in the test case definition: the development of system functions as specified in use cases, and the development of the components and systems implementing them is out of scope for the testing process definition. As the maturity of the object under test and the type of use case have strong influence on the test criteria and appropriate experiment setup, these factors are assumed to remain fixed during test and experiment specification.

The actual test system configuration and eventual lab setup, as well as the test criteria need to be refined incrementally: as more knowledge about the available infrastructure and the required testing needs, due to initial uncertainty about test system and laboratory properties, becomes available both test system and test criteria need to be adapted.

The two central aspects being refined in this specification process are therefore:

1. *Test System Configuration:* Through a process of refinement, a generic real-world scenario is first refined to the operational context of the relevant use cases, then detailed into a test system meeting a specific test objective, and finally mapped to an experiment setup in a laboratory.
2. *Test Criteria:* The test objective is broken down and refined into specific metrics. These metrics are then bound to specific parameters of a test system, quantified by means of an appropriate test and experiment design, and finally evaluated on the basis of measurements recorded in the controlled experiment.

The starting point of the envisioned procedure is the specification of a test case (i.e., Step 1), in the sense of the definition above: to identify what object should be tested within what kind of system, and to what test objective. The test case is thus derived from a scenario and corresponding system configuration as well as use cases within this setup. It aims to identify specific test criteria, relating to a test system configuration, relevant use cases and a specific test objective.

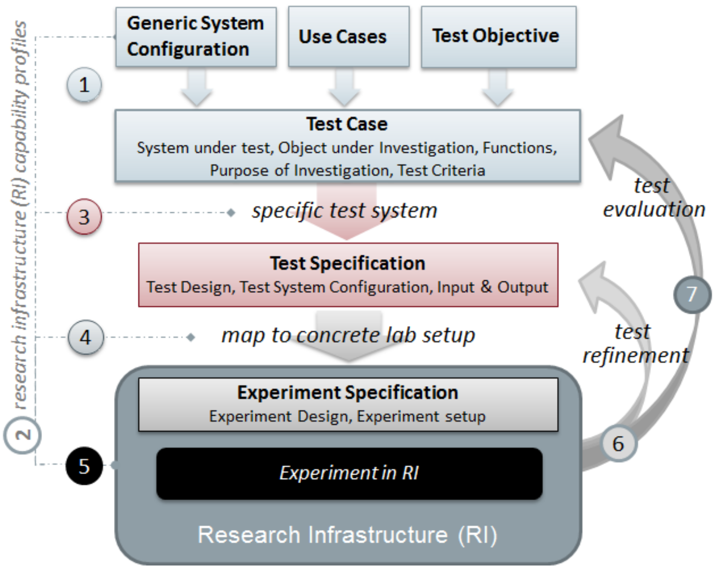


Figure 2.4: Main steps of the ERIGrid methodology, for an individual test case

To inform the specification process, the RI involved are profiled with regard to their testing capabilities, in an independent step (i.e., Step 2 in Figure 2.4). In continuation of the test case, the *test specification* (Step 3 in Figure 2.4) refines the configuration and test procedure to a concrete test system, which includes a detailed specification of the system under test and the object under investigation. Once the RI and tests are known the experiments can be specified in the sense of a detailed mapping of the test system to the available lab infrastructure (i.e. the concrete setup and design, Step 4). The experiment (i.e., Step 5) concludes the specification process. Step 6 represents the data and results collection of the experiment. Finally, Step 7 corresponds to the data analysis and combination of results to evaluate the specified test criteria.

As mentioned above, the ERIGrid approach assumes that for a holistic test it is not feasible to define and conduct a combined large-scale test incorporating all relevant domains and systems in one single setup. As illustrated in Figure 2.5, it is therefore foreseen that a *holistic test case* must be divided into sub-tests. The sub-tests concentrate on certain components, domains, or subsystems in total reflecting the structure of the holistic test in such a way that the sub-test results may be assembled to offer quantitative feedback on the holistic test criteria.

This decomposition is performed in context of the *holistic test case* and on the basis of an overall identified system under test. As a part of the mapping step (i.e., Step 3), where the interfaces and dependencies between the sub-test cases, as well as the resulting requirements, must be specified. In a second part of the mapping step, the descriptions of the sub-test cases, given the RI profiles from Step 2, are employed to identify for each sub-test case the appropriate RIs capable of conducting the test. To this end, dependencies between tests must be considered beforehand. The mapping step as well as the step of combining results of the sub-tests might be an iterative approach. Before setting up and conducting the experiments the process from holistic test to RI experiments and back should be specified as precisely as possible to minimise effort and costs.

Once the RI are known for each sub-test, the sub-test specifications can be refined to RI-specific the experiments, e.g. the concrete experiment setup and experiment procedure (i.e., Step 4). In context of carrying out the experiments (i.e., Step 5) it is necessary to analyse and to exchange data and results (i.e., Step 6) between the RIs, based on which cross-dependencies have been identified in Step 3. The results of all tests are analysed and combined to obtain the criteria with which the holistic test is evaluated (i.e., Step 7). Possible methods for combining results might be up-scaling or aggregating results.

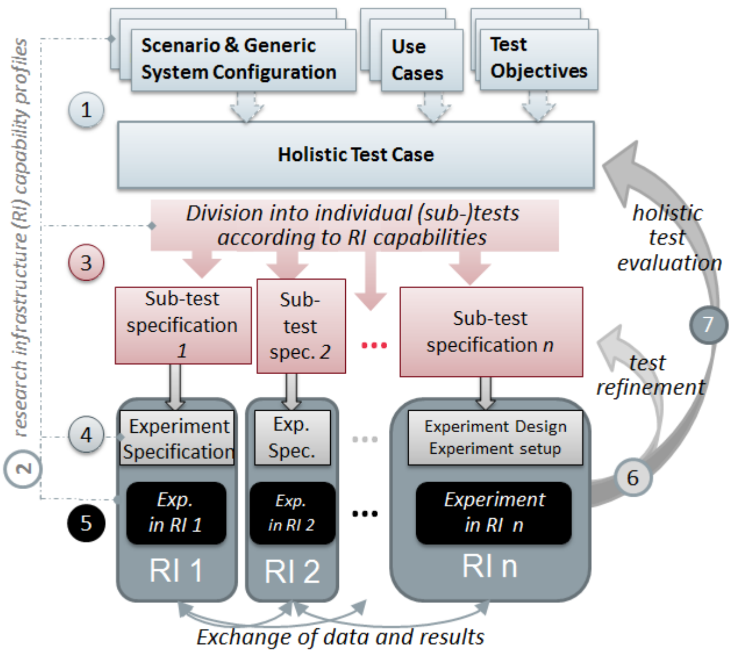


Figure 2.5: Main steps of the ERIGrid methodology applied to a 'holistic' test case, which then is divided into sub-tests to be performed at several laboratories

Thus, the mapping of a test case onto a number of sub-test specifications has two main purposes:

I. the re-use of results as an input to generate successive results, and
II. the combination of results from different sub-tests to obtain results of the holistic test.

A key component of the incremental test specification is the refinement of *test criteria*, as derived from the initial set of test objectives (purpose of investigation). However, a detailed understanding of the formulation of test criteria, the division of a holistic test case into separate cases, as well as the assessment of test results for evaluation of test criteria is beyond the scope of this report. These questions will be investigated in future work of the ERIGrid project.

2.3.3 Test Scenario Description as System Configuration, Use Cases and Test Cases

The focus of this document is to define a method for describing a holistic test scenario. As outlined earlier the proposed approach is to define a System Configuration (SC), which sets a technical context to the test; a Use Case (UC) description, which tells the overarching story of how someone interacts with a system to achieve a goal; and a Test Case (TC) description, which motivates a test or experiment and delineates test system boundaries.

The remainder of this document the focus will be on systematic test description via *test case*, *test specification* and *experiment specification*, to be presented in Section 5.

A key tool to this description is the formal method of describing system configurations that are applicable across multiple engineering domains, introduced in Section 3. These system configurations will be employed to identify the generic context and boundaries of a *system under test,* the actual *test system* and the *object under investigation*. The same description method is also used to describe available RI components and to identify their interconnections to form the experiment setup.

The introduction of use cases in Section 4 serves as reference to established specification methodology, to be used in ERIGrid context. Whereas system functions can be identified without a formal use case definition, the use case specification offers systematic requirements definition as input for the actual test case formulation.

3 System Configuration

The specification of system configurations is central to test descriptions, both to reference the system structure represented in a system test, as well as to define the configuration of the laboratory test/experiment setup for a specific experiment. The system configuration description method offers a standard way of representing systems, aimed to simplify inter-laboratory and cross-disciplinary collaboration. This section introduces the fundamental concepts and application of the ERIGrid approach to the specification of System Configurations (SCs).

Section 3.1 presents background on relevant SC description methods that relate to and inform the ERIGrid adopted approach. The fundamental SC concepts and their relation to various test description aspects is introduced in Section 3.2. Finally, Section 3.3 outlines the application of the SC method on the example of the voltage control test scenario introduced in Section 2.1.3.

3.1 Relevant System Configuration Description Methods

In this section, system modelling methods/languages are briefly introduced. These methods serve as an inspiration to the system configuration description method presented in the following section.

3.1.1 Smart Grid Architecture Model – Smart Grid Plane

The Smart Grid Architecture Model was developed by the Smart Grid Coordination Group [39], with three major objectives:

- Ensuring that the main elements of the architectural model be able to represent the Smart Grid domain in an abstract manner with all the major stakeholders. Such a model should be coherent with already existing comparable models worldwide.
- Define an architectural framework that would support a variety of different approaches corresponding to different stakeholders‘ requirements and make it in a timeframe that would force to choose a limited set of such approaches.
- Providing a methodology that would allow the users of the architectural model to apply it to a large variety of use cases so that, in particular, it would provide a guide to analyze potential implementation scenarios, identify areas of possible lack of interoperability (e.g. missing Standards), etc.

SGAM referrers to a three-dimensional view where the vertical axis (layers) refers to specification levels of ICT elements, from high-level roles and business use cases down to protocols and components. This vertical aspect of SGAM is directly related to use cases and is further discussed in Section 4.1.3.

The SGAM horizontal axes, are called *domains* and *zones*, and form together the *Smart Grid Plane* as seen in Figure 3.1. Each square of the Plane can be viewed as an area specialization within the power system, enabling reference designation of both functions and components. The concept of domains here refers to blocks in the energy conversion chain. The SGAM domains cover the full chain of conversion of electrical energy, including Bulk Generation (fossil and wind energy, nuclear and hydroelectric facilities, solar energy on a large scale), Transmission (infrastructure and organization that carries electricity over long distances), Distribution (infrastructure and organization that distributes electricity to users), Distributed Energy Resources (DER; distributed electrical resources directly connected to the public distribution system) and Customer Premises (end users and electricity producers, commercial facilities, industrial or home, photovoltaic production, storage, electric vehicles, batteries, micro turbines). The SGAM zones refer to levels of a means-ends abstraction hierarchy that characterizes the different applications of ICT elements in the power system. Here, physical components are at the Process level, process level controllers belong to the Field level, coordinating functions belong to the Station level and higher-level control room and

Operator support functions belong to the Operation level. ICT associated with Field- to Operation levels is commonly referred to as OT (operational technology), whereas ICT associated with Enterprise and Market levels is referred to as IT (information technology). IT/OT are distinguished due to their essentially different requirements priorities.

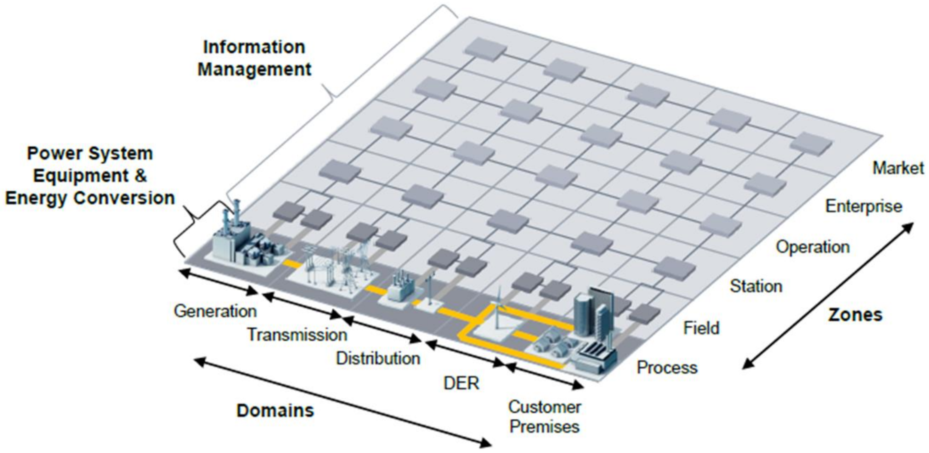


Figure 3.1: Smart Grid Plane from SGAM

The distinctions of the SGAM plane are specific to smart grid applications and offer useful semantics for reference designation. There is however no formal concept for annotation of concrete interconnections or multiplicities. As both domains and zones can define areas of specialization, either can be considered a domain or sub-domain in the sense of the definition presented in Section 2.3.

3.1.2 UML and SysML

This Unified Modelling Language (UML) [40] was created in order to aid the tasks of specifying, visualising and documenting models of software systems. It was adopted as a standard by the Object Management Group (OMG) in 1997 and accepted and approved by the International Organization for Standardization (ISO) in 2000. Despite the widespread use of UML in the modelling of software systems, the utilisation of UML is not limited to this area alone. UML was developed as successor to the Object Oriented programming concepts, which makes it flexible enough for modelling other system types emanating from the real world, for example manufacturing processes.

UML has many advantages, as it was developed with the intention of coping with large enterprise applications whereby the typical challenges are driven by issues of complexity, scalability, security, and robustness. In particular, the level of abstraction offered by UML, allows the user to focus on modelling the different aspects of the system during the design and development phase, but also without having a bearing on the actual analysis or design methodology utilised in the construction of the system. It is especially beneficial in multidisciplinary fields as it provides standardised modelling terminology, as well as standard diagram types for visualisation. This ultimately leads to improved communication and management of system complexity. A well-known benefit of UML is re-use of information and data. As the system grows, it is possible to keep a library of model of components which can be reused at a later date, resulting in faster system development times.

To overcome some limitations of UML in the for applications in systems engineering, the Systems Modeling Language (SysML) was introduced as a general purpose visual modeling language. SysML is defined as an extension of a subset of the Unified Modeling Language (UML) using UML's profile mechanism[8] “and supports the specification, analysis, design, verification and validation

[8] Source: https://en.wikipedia.org/wiki/Systems_Modeling_Language

of a broad range of systems and systems-of-systems” [41]. SysML expands the UML 2 standard diagrams with two new types (requirements and parametric diagrams).

Both UML and SysML are formally specified and the standards are supplied with a syntax to facilitate translation between graphical and machine-readable textual representations (e.g. in XML). The SysML concepts of system and sub-systems modeled as objects are relevant to ERIGrid.

3.1.3 The Common Information Model for Power Systems (CIM)

The deregulation of the power infrastructure has increased the need for Transmission System Operators, Distribution System Operators and Utilities to communicate for the purpose of planning and operation of the power system. Specifically, it has increased the need for information modelling with respect to the power system. This has led to the definition of the Common Information Model (CIM) in standards IEC 61970 for transmission systems and IEC 61968 for distribution systems. The CIM is “based on a Unified Modeling Language (UML) based information model representing real-world objects and information entities exchanged within the value chain of the electric power industry” [42] therefore it uses the description language used for object-oriented software architectures, as seen in Figure 3.2; CIM is therefore language independent.

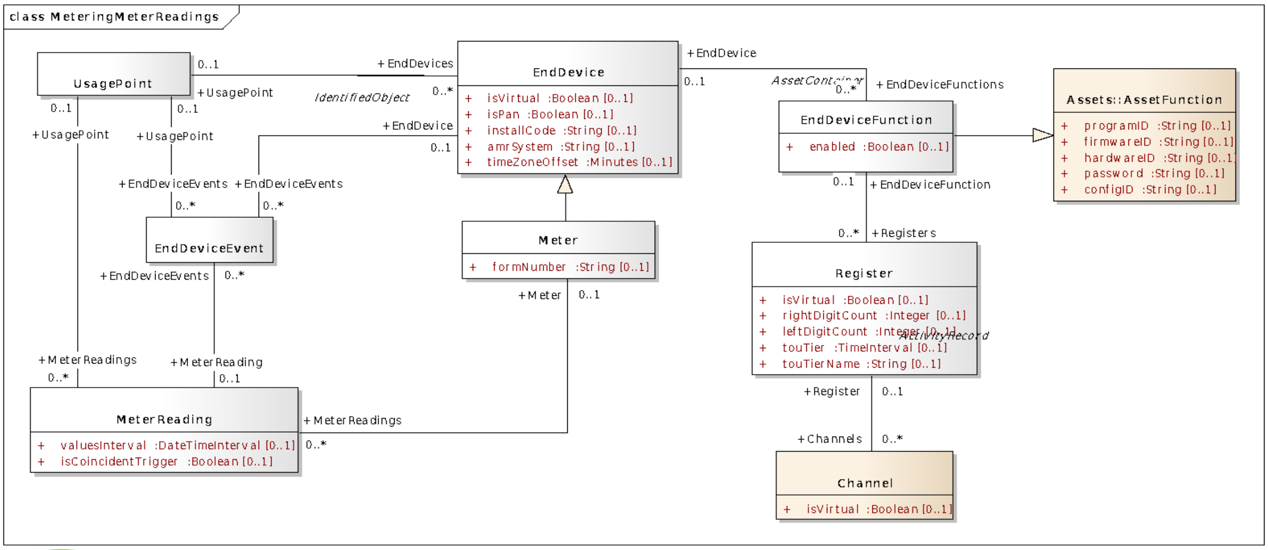


Figure 3.2: Example of CIM for metering and control [42]

CIM is organized in packages, each containing a set of classes with their structure, attributes and associations. CIM defines a common vocabulary and ontology for the electric power industry. It is mainly used in data exchange for EMS applications and energy markets.

Figure 3.4 gives a general view of the application domains of CIM with regard to other popular information models. While IEC 61850 focuses on station and field level, mainly on communication within substation; both Multispeak and CIM focus on interfaces between applications above station level. Whereas the main interest of Multispeak is the distribution domain, CIM covers part of generation, transmission, distribution and DER domains. The generation domain is not fully covered which is the reason for the partial coverage shown in Figure 3.3. In general, CIM is used for two primary objectives:

- *Exchanging data between applications:* In this case, the messages use CIM Semantic and are formulated into XML serialization.
- *Encapsulating entire power system models:* In case of exchanging topology data of the system or of networks, the XML hierarchy becomes insufficient. The Resource Description Framework

(RDF) is an XML schema that provides the possibility to define other relationships between XML nodes. The combination XML/RDF allows a set of objects to be expressed as XML while retaining their relationships and class hierarchy.

CIM/XML/RDF ensures the possibility to exchange static and dynamic data as well as the current state of electrical networks in a standardized way, which leads to a seamless semantic data exchange among components in a platform and among partners in the working network.

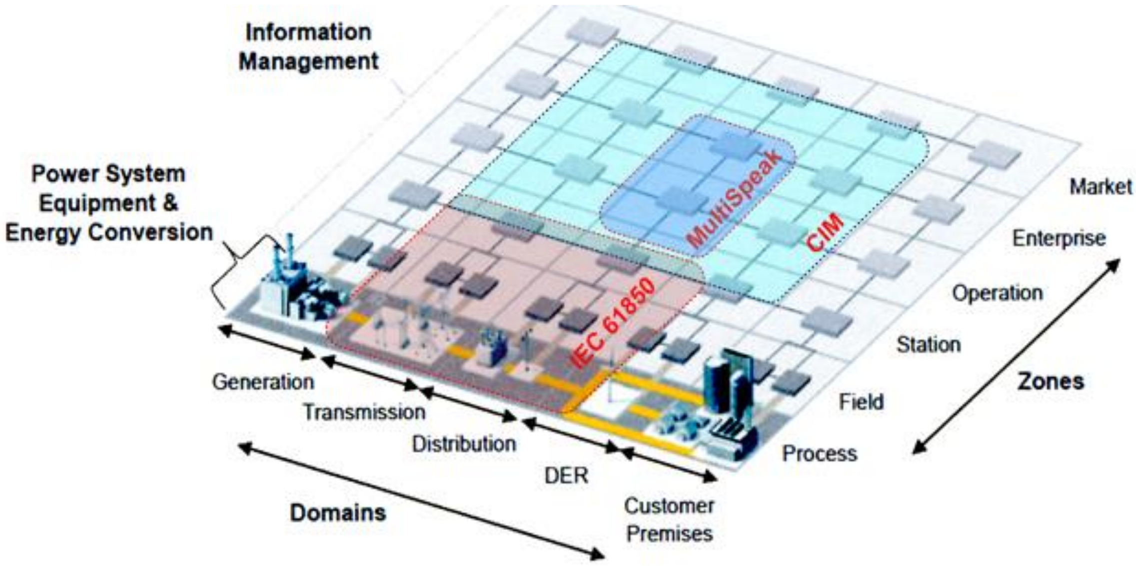


Figure 3.3: A comparison of information domains on the smart grid plane of SGAM model [43]

CIM is a platform and transport independent model, and allows a detailed and extensible modeling for storage or data exchange. In order to successfully apply CIM, it is necessary to setup a suitable communication protocol. The method with which to build a database from CIM depends on several factors. The structure of the database varies from one system to another. In particular:

- A CIM database is derived from a CIM and is capable of storing data defined by CIM in a structured way.
- You can import and export data from a CIM structure from the database.
- There is no standard for IEC CIM database, however, interfaces to import and export are generally standardized

The IEC 61968 standard specifies the use of XML (eXtensible Markup Language) and RDF (Resource Description Framework) to symbolize the CIM elements. XML is used as message format in IEC 61850, CIM and Multispeak. XML is a meta-language that allows the description of data structure. In XML, the data is encoded as plain text and is platform independent. However, a basic XML document cannot denote any link between two elements that is not inheritance relation. RDF is an XML schema that brings the notion of property, a link between the objects other than hereditary links with the concept of URI (Uniform Resource Identifier) among others. The combination of CIM / XML / RDF / RDF Schema provides a full representation and object oriented electrical system as text, standardized, independent of platforms and extensible (Table 3.1). This allows informative, easy and efficient communication between system components and between systems.

Table 3.1: Modeling example of an element with CIM RDF XML

Element	CIM RDF XML
VoltageLevel Name : VLA highVoltageLimit : 35.0 lowVoltageLimit : 31.0	<cim:VoltageLevel rdf:ID="_xyz987654321"> <cim:IdentifiedObject.name> VLA </cim:IdentifiedObject.name> <cim:VoltageLevel.highVoltageLevel> 35.0 </cim:VoltageLevel.highVoltageLevel> <cim:VoltageLevel.lowVoltageLevel> 31.0 </cim:VoltageLevel.lowVoltageLevel> <cim:VoltageLevel.BaseVoltage rdf:resource="#_jkl567890"/> </cim:VoltageLevel>

3.1.3.1 Connectivity in CIM

In CIM, connections between elements are represented by each element having a *Terminal* which connects to a *ConnectivityNode [44]*, see Figure 3.4.

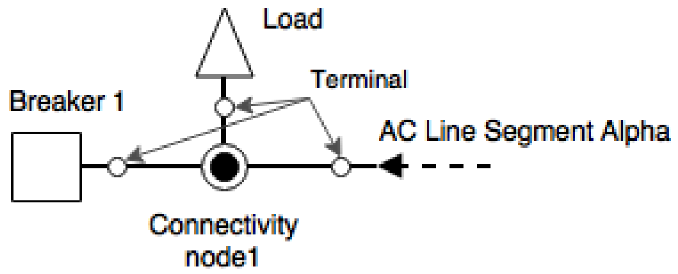


Figure 3.4: An example circuit of how connectivity is represented in CIM [3]

Each element may have 0 or more *Terminals*. To express a connectivity, each *Terminal* connects to a single *ConnectivityNode*. A *ConnectivityNode* may be connected to 1 or more *Terminals* and all elements whose *Terminals* are connected to the same *ConnectivityNode* are interconnected. In CIM, the concept of *Terminals* helps defining the points of connectivity related measurements, such as current flows and voltages. An example of a transformer representation in CIM can be seen in Figure 3.5, where the two terminals connect to different components of the electrical system.

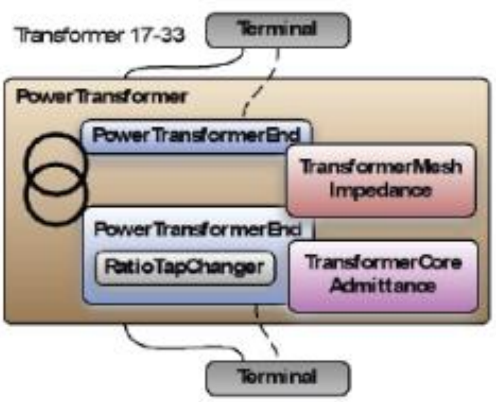


Figure 3.5: An example of a transformer representation in CIM [3]

3.1.4 IEC 61850 Substation Configuration Description Language (SCL)

The IEC 61850 standard is substation automation and communication standard. IEC 61850 SCL (Substation Configuration Description Language) specifies how configuration of communication in electrical substations, with a focus on intelligent electronic devices involved in communication, is defined. The standard makes it possible to:

- Define communication objects and methods
- Include object model (standard / extensible)
- Standardize the configuration language
- Specify compliance testing

In addition to specifying communication parameters of both the overall communication system and devices, this standard goes a long way in also providing a method to describe switchyard (function) structures and define the relationships between them. Model description can be carried out using UML, while the configuration language utilises XML(Extensible Markup Language). The actual implementation of individual entities is not constrained as the language only deals with configuration. The abstract data models defined in IEC 61850 can be mapped to a number of protocols, including MMS (Manufacturing Message Specification), GOOSE (Generic Object Oriented Substation Event), SMV (Sampled Measured Values) and Web services. These protocols can operate over TCP / IP or high-speed substation local area networks using switched Ethernet to achieve the required response times of less than four milliseconds for protection relays.

3.1.5 Relevance and Conclusion

The system configuration description method for ERIGrid requires a simple domain-independent approach of representing systems. Each of the above mentioned system modelling methods/languages covers a relevant aspect of the system description, but none of them is able to model all the system characteristics that are relevant to ERIGrid. SGAM offers semantics for smart grid domains, but cannot provide a configuration language. UML and SysML offer generic and computer-readable system description methods that could be adopted as configuration languages, but also entail a complex toolset. CIM and IEC61850 SCL each offer flexible connectivity notions and are well supported in the power systems domains. What limits their adoption in the context of research infrastructure is a) the lack of a multi-domain approach (domain-independent system modelling), and b) the need for light-weight descriptions in a lab context, which is relevant in particular at early development and prototyping stages where full standard adoption is not yet required.

3.2 The ERIGrid Approach to Description of System Configurations

The system configuration (SC) description is meant as a generic representation method for systems, to facilitate exchange of specifications across disciplines and laboratory infrastructures, while offering a shared and re-usable method of specification that is compatible with existing approaches. The challenge is therefore not to define an entirely new approach, but one that is practical and aligns well with existing approaches that were defined for purposes other than test description. Similar to other related specification work such as Smart Grid use cases reference designation in the Smart Grid Architecture Model (SGAM), information modeling for power system ICT via the Common Information Model (CIM), or other applications of the Unified Modeling Language (UML) or the Systems Modeling Language (SysML)). In contrast to CIM, a multi-disciplinary approach is required, as test specifications may involve non-power systems components, and domain-specific notions from ICT or power systems domains should be accommodated. Another requirement is the adoption of near-standard approaches so that later mapping to standards is feasible and deeper model conflicts can be avoided. Finally, the description method should be flexible to accommodate different levels of detail in the description, as well as different contexts, according to the different contexts of specification that are part of the ERIGrid test scenario description, while maintaining an alignment of the defined systems and components:

- For *Use Cases* the use of a SC description is similar to SGAM domains/zones, offering a principal context and abstract system boundaries for the definition and allocation of functions in a system architecture; systems are not concrete, and the context is independent of eventual test specifications
- For a *Test Case* the SC description is also generic, but in contrast to the use cases, system boundaries now are specific to a testing context, component types relate with specific functions and connection types and domains under test need to be identified; further, an overlay identifying systems, components and domains under test.
- The Test System, as SC description for a *Test Specification* is again more specific, and can be treated as an instance of the test case SC; the system under test (SuT) and an object under investigation (OuI) for are uniquely identified and each system, component, and connection is labelled and uniquely identified.
- In this sequence, the Experiment Setup for *Experiment Specification* is the first SC to address lab components and their connections. In principle, only the OuI part of the SuT has to be accommodated explicitly, defining the coupling btw. OuI and research infrastructure; the remaining SuT components may be represented in other ways.

To enable the experiment specification based on a research infrastructure (RI) database, two further SC types are needed, offering a role complementary to the generic SC descriptions above:

- The RI description, as an entry in the RI database defines the concrete components available in an RI, including potential multiplicity and potential connectivity; it is therefore specific in that it defines concrete instances, but generic in that it does not define all connectivity.
- Finally, and RI information model is required that defines the types of components and domains that may be included in an RI for *RI profiling*.

The description method described below, has the above requirements in scope. The description method should thus be able to represent all features of the multi-domain systems relevant to ERIGrid. There is no further discussion of computer-readable formats in the following, but due to the adopted formats, the assumption holds that computer-readable exchange formats can easily be formulated on the basis of the given formal structure. The application of SC description to the test case SC, test system, and experiment setup is demonstrated below. The development of a RI database and the filling of RI database entries will be reported in the ERIGrid Deliverable *D-NA5.2: "Partner profiles"*.

To integrate the SC description methods used in smart grid disciplines (e.g. electrical, ICT, or thermal systems), ERIGrid adopts and generalizes basic system description concepts that are employed in the power system CIM.In particular, the concept of domains is adopted from SGAM, and SysML provides the concepts of system and sub-systems modeled as objects. The concepts of Terminal and ConnectivityNode are adopted from CIM and extended also to other domains than the electrical one.

Section 3.2.1 introduces the basic description concepts; the following Section 3.2.2 outlines how these concepts are interpreted to form SCs for the different applications. Section 3.2.3 then outlines different variants of data structures for SC representation.

3.2.1 System Configuration Concepts

A system configuration has been defined in ERIGrid to include *domains*, *components*, *connectivity*, *constraints* and *attributes*. The formulation of these description concepts occurred in part on a bottom-up basis, formulating the description needs of ERIGrid D-JRA1.1, which defined several "Generic System Configurations" to outline the scope of ERIGrid test scenarios and use cases. The description has been necessarily generic and somewhat informal.

To cast these concepts into a formal structure, an upper ontology model has been identified, which is illustrated in Figure 3.6, and can be described as follows: *System*s are composed of *Components* and are themselves components. Components have *Terminals*, which may have directionality and are associated with one *Domain*. Domains can be structured hierarchically. Two or more terminals associated with the same domain can be connected using a *Connection Point*. Attributes All of the above are *System Configuration Objects*. *Constraints* can be associated with any type of system configuration object. A set of them composes a *System Configuration Container*, which has a system configuration type (*SCType*) attribute.

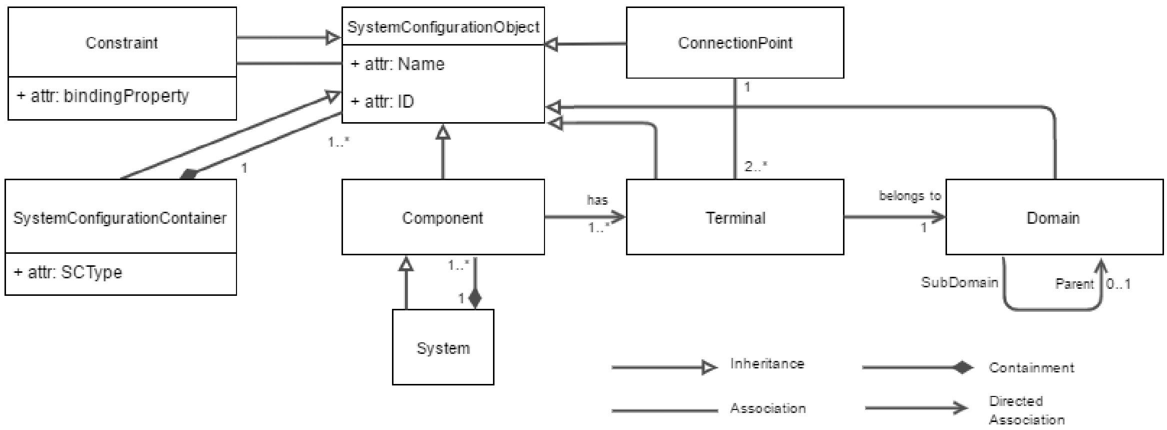


Figure 3.6: System Configuration concepts

Given this ontological, or object-oriented approach to system modelling, there are two overall types of specificity, corresponding to the concept of “domain model” (information model) and “instance model” in ontology modeling:

- *Generic System Configuration (GSC):* a description of the types of classes that can be part of a system configuration, establishes “semantics” of concepts to be employed in specifying a SC.
- *Specific System Configuration ((S)SC):* correspondingly, an ‘instance’ of a GSC, representing specific objects, such as concrete lab component, or specific connection between components.

Each of these concepts is discussed in the following.

3.2.1.1 Components and Systems

Components are the items that a system is eventually composed of. The type of components varies a lot depending on the domain and the actual function of the component. Components can be practical technical devices, but they can also be more abstract entities or subsystems. Common to systems and components is that a clear *system boundary* can be identified. The distinction of system and component is a question of the frame of reference: for example, a “component” such as a DER unit can also be viewed as a “system” composed of e.g. a physical energy conversion device and an ICT-based built-in controller.

3.2.1.2 Domains

Domains can refer to infrastructure-specific operation areas such as electricity, heat, primary energy resources or ICT. Within the ICT domain, one may identify communication protocols as a subdomain, and within that, specific protocols (e.g. TCP/IP) can be further identified. Fundamentally, the concept is defined as introduced earlier:

> A *domain* is an area of knowledge or activity characterized by a set of concepts and terminology understood by practitioners in that area.

The ERIGrid modeling approach should be generic and transferrable to any of these “domains”, so there is no pre-existing firm taxonomy of domains. This generic definition indicates the expertise associated with a domain.

In many smart grid development areas domains become more interlinked and can be perceived as overlapping, creating an increasing number of ‘hybrid’ domain, as a component that converts between electricity and heat then would be associated with each of these domains. Instead of defining hybrid- and multi-domain components, such a component is said to have terminals in each of the associated domains. Components can therefore act as an interface between several domains. On the other hand, connectivity between components should be specific so that interoperability between systems and components can be specified as a subject requiring a particular form of expertise. Finally, it is apparent that domains can follow a taxonomy of sub-domains (e.g. refining ICT-domain connections to a particular protocol) in which the connectivity is categorized on a more detailed level. The more refined and specific the problem, the more refined the domains of a test system may become.

3.2.1.3 Connectivity: Terminals, Domains, Connection Points

Connectivity defines how and where components are connected. The approach used to express connectivity in CIM expresses connectivity between components in two steps: i) as terminals associated with a component, and ii) by connectivity nodes associated with terminals, expressing an actual interconnection of components. This approach has been adopted and extended to a more generic concept by association of terminals with domains.

In a specific SC, terminals can thus only be connected if these are associated with the same domain. A specific connection between components is thus expressed by associating the same connection point with two or more terminals from the respective components. If a domain-hierarchy is available, terminals can be connected with terminals from parent-domains.

In a generic system configuration, direct connections are not required, so that connectivity can therefore also be expressed as a ‘potential connectivity’, or further even express a semantic relation between components of a system configuration that does not warrant a direct connection, such as the concept that a “state estimation” is a representation of a specific part of an actual distribution grid. In particular, in case of generic system configurations, there is a need to express this type of ‘relational’ or ‘abstract’ connectivity.

3.2.1.4 Attributes

Attributes define the characteristics of system configuration objects. Practically, there are two attribute types: Global attributes include some information on prevailing circumstances which is common to multiple components, for instance outdoor air temperature. In contrast, component attributes are specific to certain components and can be very detailed attributes which cannot be applied in other components. For instance, DER unit nameplate details are obviously component attributes. Attribute types (as well as ranges or default values) are defined in a generic system configuration; the types may be amended in a specific SC.

3.2.1.5 Constraints

Constraints describe limitations to component or system functionality or behaviour. Constraints can be caused by operational circumstances (for instance a regulatory framework), technical limits (for instance voltage, frequency), prevailing legislation or rules (for instance grid codes), dependencies from other components (for instance availability of communication connection), interoperability (for instance access to right format data) or other practical issues that can limit operation. A constraint is bound to a specific scope, such as a combination of specific attributes, but it may address e.g. all of a specific type of component, a specific component, or it may affect all components that are part of a system configuration. So-called global constraints can be associated with a System Configuration Container, and apply to all elements of this particular system configuration.

Constraints have not formally been addressed in the SC description method. The constraint attribute “bindingProperty” is used as a placeholder to formulate the constraint property limiting the associated System Configuration Objects. Formal constraint description mechanisms, such as the Object Constraint Language (OCL) [45] may be applicable to this end.

3.2.2 Applications of System Configuration Description in ERIGrid context

Based on the description concepts introduced above and the overall outline on the holistic test specification procedure, system configurations can be required in a number of different forms and contexts.

Intended Uses of ERIGrid SC description

In application to the ERIGrid test specification procedure, there are lab- and real-world oriented system specifications contexts:

- The “real-world” context expresses concepts that need to be analysed and finally to be represented in an experiment;
  - Incremental refinement here leads from the generic reference designation required for use case and function specifications to more context-aware formulation of a generic system configuration for a test case
  - The test system finally expresses the exact subject of a test specification.
- The “lab” context describing the configuration of a research infrastructure (RI),
  - In generic sense, data models for types of lab components must be formulated;
  - A concrete laboratory is to be represented with its potential connectivity.
- In the final “experiment” context,
  - the relevant parts of a “real-world” configuration are represented by the lab infrastructure (expressed by a mapping between real-world test system and the lab components configuration)
  - the actual object under test is integrated in the lab setup (E-SC).

The relevant types of representations of SCs are thus summarized in Table 3.2.

Table 3.2: Classification of System Configuration Types (SCTypes)

Name/ Purpose	Context	GSC/ (S)SC	SCType	Explanation
Function-System Alignment	Use Case	GSC	UC-GSC	As SGAM domains & zones: reference designation for functions, independent of test case. Corresponds to D-JRA1.1 Generic System Configurations.
Test Case context model	Test Case	GSC	TC-GSC	Establishes type conventions for test case: relevant SC component types, domains, etc., and categorically identifies the SuT (and optional OuIs); specifies multiplicities; “class model”.
Test System	Test Specification	(S)SC	TS-SC	A concrete instance of TC-GSC to address a specific OuI and test criteria; labelled terminals and specific connections; OuI and SuT identified as overlay annotation.

Name/ Purpose	Context	GSC/ (S)SC	SCType	Explanation
Experiment Setup	Experiment Specifica-tion	(S)SC	E-SC	The configuration and interconnection of RI compo-nents, representing the SuT, and including OuI; al-so “Test Setup”
RI Description	RI database entry	(S)SC	RI-SC	Lab configuration with components, including po-tential multiplicity and potential connectivity of lab components, but may have undefined connectivity.
RI information model	RI profiling	GSC	RI-GSC	Specification of Lab profiling data structures, includ-ing component types and domain types.

To be able to conduct a test, the test specification has to be mapped to a lab configuration. The TS-SC presents the components that are needed for the test case in a real environment whereas E-SC represents the actual test/experiment setup in a particular lab. For instance, electrical net-work can be represented in an E-SC using a real time simulator whereas in a TS-SC it is repre-sented using line segments, transformers, loads etc. Examples on constructing TC-GSC, TS-SCs and E-SCs are given in Section 3.3.

Related work on SC descriptions within ERIGrid

The fundamental description notions were agreed in a joint technical workshop in the beginning of the project, asserting that:

- System configurations are a concretization of scenarios relevant for test description; the term scenario would be used to refer to a context of a more general character
- Component-centric description (inspired by the object-oriented architectures)
- Sub-systems can be viewed as components
- Components are found within- or on the border- of domains (the notion of “components as in-terfaces between domains” is a logical clarification on the SGAM domains)
- Any object of an SC can have attributes (inspired from object-oriented architectures)
- Constraints can be applied to components, systems and domains

The description strategy and relation to the state of the art, as well as the various graphical and tabular annotations were developed later. The first applications of the system configuration de-scription method were in D-JRA1.1 and where examples of generic system configurations were composed. The generic system configurations of JRA1.1 provide an abstract description of three selected system areas: distribution grid, transmission grid as well as offshore wind and vertical in-tegration. The generic system configurations include information on domains, components, con-nectivity, constraints, attributes, associated use cases and reference to high-level scenarios. These generic system configurations provided a context for the follow-up specification of ERIGrid focal use cases and holistic test cases reported in D-JRA1.2 and D-JRA1.3, respectively. Relevant sys-tem configuration descriptions include also configurations of pure co-simulation setups discussed in JRA2 and real-time PHIL/CHIL setups addressed in JRA3.

3.2.3 SC Data Structure and Description Variants

The fundamental elements of the system configuration description have been presented in Section 3.2.1, and summarized in Figure 3.6. A SC based on these concepts can be reported in different forms, including a document form, tables, diagrams, or computer readable formal structures and databases. In this section we focus on introducing a table format and a domain-independent graph-ical notation, which are meant to be used in ERIGrid documentation.

The main structures of a table- based definition are listed as follows. The corresponding graphical notation is exemplified afterwards. Note that corresponding illustrations may be provided on the basis of domain-specific diagrams (e.g. a one-line diagram for Power systems, or a PI-diagram (piping and instrumentation) for fluidic processes). As a domain-specific diagram typically represents a domain-specific aspect of a system configuration, a holistic system configuration must allow for several domains to be combined. For sub-systems that can be entirely specified in a single domain (e.g. the electrical topology of a *distribution network*), it is recommended to retain the domain-specific representations where possible. An aggregated view of the domain-specific subsystem (component: "distribution network') will then be included in the multi-domain SC, with all connections across the sub-system boundary (e.g. to loads or transmission system) expressed as terminals (e.g. low-voltage terminals).

In summary, a textual description of a System Configuration includes:

- A *component table* that clusters components by system/sub-system and denotes any specific attributes of the components that are not bound to component terminals in the same domain. Components are assigned a component ID.
- A *connectivity table*, which lists the component IDs and denotes which kind of connection it has in a specific domain (in a generic SC), or a connectivity table, which lists the component IDs and denotes in which domains the component has terminals (in a specific SC).
- A *topology table*, which lists the terminals connected to each connectivity node (in a specific SC).
- An *attribute table* that describes the common attributes of components within the defined domains as well as their corresponding constraint.
- A *constraints table* listing attributes and constraints at different scenario levels.

3.2.3.1 Components and Systems with Attributes

As systems are composed of components and can be divided into sub-systems, there can be a hierarchical annotation. In table form, this can be annotated by grouping components into systems and sub-systems. Attributes are specified for components, but if an attribute is specified for a system, it must be applicable to all its components.

Table 3.3: Template for Component and System description table

System	Sub-system	Component	Attributes
system-name	sub-system-name	component-name	

3.2.3.2 Domains

Domains have the role to host connectivity across components, and are viewed as areas of expertise. As domains can be cascaded hierarchically, a sub-domain is also a domain, just associated with a more specific context. E.g. a domain may be ICT, and a sub-domain may be the TCP/IP protocol. The more specific a domain is defined the more realistic is that a specified connectivity is realizable in practice. Ideally, a common domain taxonomy is used across the project.

At least within one test description context, the same domain type hierarchy (DTH) is to be used, and may be extended with refinements where necessary. Figure 3.7 presents a taxonomy to identify the different ways in which connectivity can be specified. The domains with icons have been adopted from the domains introduced in D-JRA1.2. This taxonomy can be used as starting point for a test case domain type hierarchy. Note that subtle but significant changes in the modelling assumptions must be observed, when connectivity is shifted between different branches of domain categories. For example, in shifting from a) *Functional->Control* to b) *Signal->Continuous* the reference components that are connected is also shifted in a) the sending and receiving components are of a functional nature and not a concrete entity, whereas b) assumes a concrete entity such as a PLC that might actually *host* the function referred in case a).

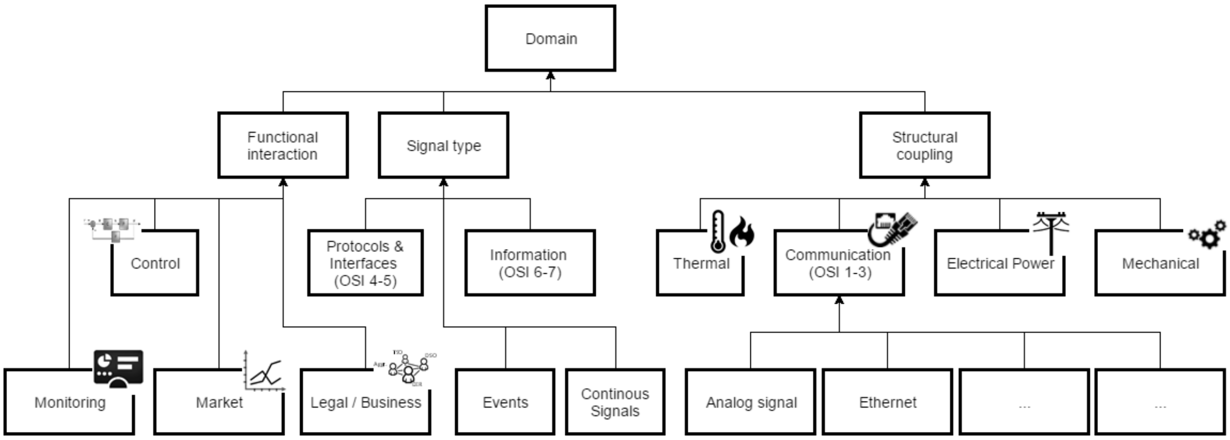


Figure 3.7: Taxonomy of Domains modelled after D-JRA1.2 Focal Use case domains

As noted above, *terminals* are bound to a domain. It is therefore meaningful in a table-based specification to define the terminal types and their properties directly on the basis of respective domains.

Table 3.4: Domains and Terminal types

Domain	Sub-Domain	Terminal Types
Domain-name	Sub-domain	e.g. directed on undirected

3.2.3.3 Connectivity Annotations

The most direct and intuitive expression of connectivity is done using a graphical representation. To that end, a number of figure types have been defined:

- Multi-domain “Generic” System configuration
  - Connectivity is expressed using the concepts of terminals and domains; all components and connections are meant as generic “classes” defining connection types
- Multi-domain “Specific” System configuration
  - The components and their connections represent specific labelled “instances”
- Intra-domain or “domain-specific” representations of connectivity
  - In case a whole set of components and their connections belong to a single domain and a well-accepted conventional domain-specific view is available that offers a full connectivity specification, it can be employed to represent a complete *sub-system* of a multi-domain system configuration.
  - In the ERIGrid methodology, domain-specific representations should be complemented with a multi-domain SC, which clearly identifies the subsystem boundaries and its labelled external terminals.

An example of this use is presented in the combination of Figure 3.12 and Figure 3.13, where a one-line diagram represents the power grid topology, which is modelled as “distribution grid” system in a related multi-domain diagram, exposing electric and informational terminals. In principle the figures described above correspond to data represented in tables: also domain-specific annotations (e.g. electrical grid topology data) can be interpreted in terms of the multi-domain figure notation (graphically expressed in Figure 3.11). These can either be separately drawn or automatically composed from the connectivity and topology table data.

Component Terminals

If a component is associated with several domains, it has at least one terminal for each domain. Figure 3.8 illustrates the graphical notation. Table 3.5 outlines that in a specific SC these terminals have to be identified and labelled, whereas in a GSC the domain-specification is sufficient (c.f. example in Appendix, Table 9.3).

Table 3.5: A component-domain table defines in which domains a component has terminals

Component	Domain	Terminals
Component-name	Domain or Sub-domain	Terminal-name (terminal-type)

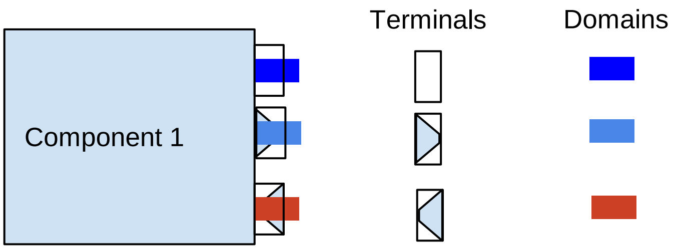


Figure 3.8: Example component with terminals in three domains.
The uppermost terminal is bidirectional and the following two are directional

Connection Points: Topology in a Specific System Configuration

A *connection point* is used to express connection between terminals within the same domain. Connections in a specific system configuration must be expressed by connectivity. Connections cannot be expressed across domains. The above *connectivity table* maps component terminals to domains. The actual system *topology table* lists the terminals connected to each *connection point*. The graphical notation is presented in Figure 3.9. Note that by convention, the connection point illustration is omitted if only two terminals are connected. Naturally, there have to be domain-specific rules pertaining the aspects of directionality and multiplicity of connections.

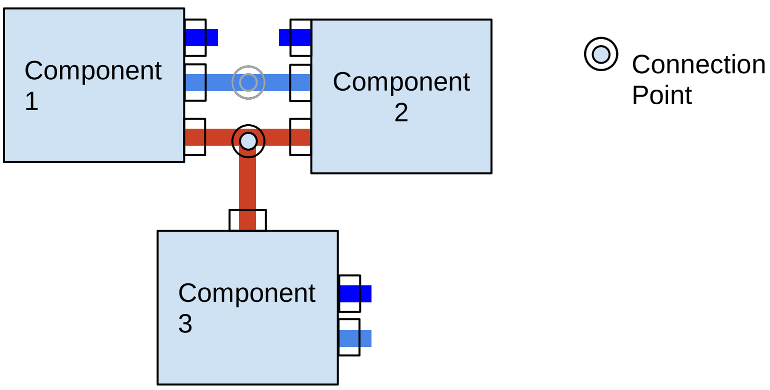


Figure 3.9: Connection Points are used to connect terminals of different components

The topology table lists the terminals connected to each connectivity node. The topology table is most suitable for automated usage e.g. in a database. For human users, the graphical representation is more intuitive.

Table 3.6: Topology table defining actual connections

Connectivity node	Terminals connected to the connectivity node

3.2.3.4 Attributes

As attributes are specific to components, they can also be defined to types of components to be inherited by all components of this type. There is no preferred graphical annotation for attributes, even though annotations from UML may be adopted. Any object-oriented annotation convention is suitable. The representation form for component attributed of Table 3.7Table 3.8 is a best practice adopted from D-JRA1.1, and in the Appendix, Table 9.4 provides an example.

Table 3.7: Attribute table

Type of Component	Type of Attribute	Additional info

3.2.3.5 Constraints

Constraints can be associated with any components, domains or be expressed otherwise across these limitations. They typically refer to limitations on attributes. There is no defined method for illustration. Conventions for the annotation of constraints may be taken e.g. from SysML. The representation form of Table 3.8 is a best practice adopted from D-JRA1.1, and is exemplified in Table 3.10.

Table 3.8: Constraints table

Global constraints (domain related)		
Domain	Constraint	Additional info
Specific constraints (related to Component)		
Component	Constraint	Additional info
Indirect constraints		
Cause	Constraint	Additional info

3.2.3.6 Abstract Connectivity types applicable to Generic System Configuration

In Generic System Configurations, connectivity types between different component types are presented whereas in Specific System Configurations connections between actual component instances are represented. Furthermore, for a more general context where perhaps business stakeholders or more abstract contextual relations and roles need to be expressed in a Generic System Configuration, abstract connectivity types have been introduced. The types defined in Table 3.9 have been used in context of D-JRA1.1 and their use is illustrated in Annex Section 9.3.2, Table 9:2 and Table 9.3.

For the purpose describing a specific system configuration, the abstract inter-domain connectivity, such as “AD” and “IP” cannot be expressed in terms of abstract connectivity, but rather will be refined as a sequence of direct connectivity via interfacing components. Such indirect connectivity is also relevant for the stakeholder relations. Instead of affecting the actual system configuration, they relate to the interpretation of performance indicators and functions, which are part of a test case, but are not expressed in the system configuration.

Table 3.9: Abstract Connectivity types.

Connectivity type	Explanation
DP	Direct Physical coupling (intra-domain)
IP	Indirect Physical coupling (either mediated, e.g. by a power converter by other technique; also applicable to 'equivalenced' components)
DD	Direct Data: direct field-related data for real-time control & decision purposes; e.g. as recorded in the field, is transferred from/to this component
AD	Abstract Data, such as aggregated or stored field data or otherwise abstracted and data, such as configuration data: only highly processed information is transferred from/to this component/domain
ICC	Information, processing, or Communication Container: as processing or communication function, no relevance of information content
(R)esponsible	Stakeholder is responsible for Domain/Component
(D)irective	Stakeholder directs Components or other Stakeholders
(O)wnership	S. owns component
(OP)erates	S. operates component
(T)ransactive	S. executes transactions with respect to component/domain
(I)nformational	S. acquires information from
(M)anufactures	S. produces component or system

3.3 Examples for Illustration of System Configuration Description

The following example presents the formation of real-world and laboratory system configurations for centralized voltage control application. The system configurations are loosely related to the “coordinated voltage control” use case and the corresponding PHIL/CHIL test setup introduced above in Sections 2.1.1 and 2.1.3, respectively. The system configurations are therefore related to the “Distribution grid” generic system configuration presented in D-JRA1.1 and the generic system configuration is not repeated here.

First the test case context model will be introduced (TC-GSC), then several variants of the test system configurations related to the test specification (type TS-SC) are illustrated. Section 3.3.3 introduces the experiment setup for experiment specification (type E-SC) which describes how the laboratory infrastructure and Object under Investigation (OuI) is used in the test. The tutorial does not address the RI related system configurations (RI-GSC and RI-SC), as these will be covered in D-NA5.2.

3.3.1 Test Case Perspective

The real-world system configuration is related to the use case definition and describes the real distribution network implementation of the use case (type UC-GSC). As the use case is modeled using SGAM in Annex 9.4, it will not be detailed here. For the test case, the context is further refined to highlight the elements of that generic system configuration which need to be represented in a test system to reflect the relevant operational context.

Consider the following use case aspects. A central controller is installed at substation level and is initialized with all the necessary static data of the network that it will control: network topology, admittance of lines and transformer, nominal power of DER units and storage systems, operating limits of DER units, storage systems and OLTC. While it operates, it requests and receives real-time power measurements from the smart meters of loads and DER units, as well as the state of charge (SOC) of the storage systems and the current tap position of the OLTC, in discrete iterations (e.g. every 10 minutes). Using this dynamic data, it formulates an optimal power flow problem, whose objective function involves the minimization of voltage deviation of critical nodes from the nominal value, power losses of the lines and transformer, and tap change operations of the OLTC. The outputs that result from the solution of this optimization are set-points for all controllable devices located in the network. Specifically, the controller calculates a set-point for the tap position of the OLTC, reactive power set-points for the inverters of DER units, as well as active and reactive power set-points for the storage systems. The reception of measurements and transmission of set-points is carried out through a communication network.

In a graphical representation we therefore capture the main aspects that define the distribution grid (with OLTC), voltage controller and the inverters addressed above. Further, as relevant aspects, the electrical distribution level connections, and the communication between physical infrastructure and voltage controller are named, calling for representation of electric power and ICT domains. The Generic System Configuration presented in Figure 3.10 illustrates these elements.

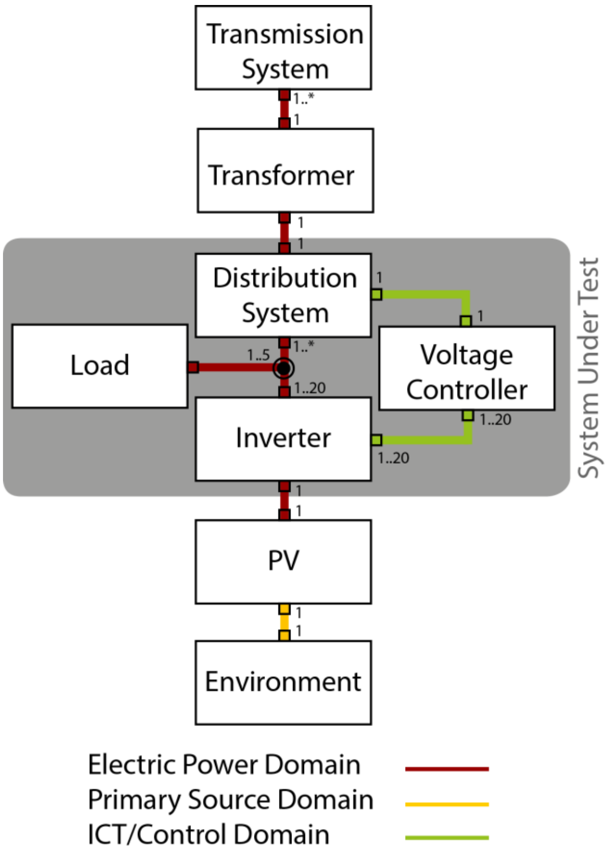


Figure 3.10: Graphical representation of the TC-GSC for the coordinated voltage control test case

Further details of this test case generic system configuration are omitted here for brevity, but are reported in Annex 9.3. As an exception, we demonstrate the annotation of generic constraints for this test case in Table 3.10. The constraints listed here, indicate how important the listing of constraints, even if qualitative, is for the annotation of a test case.

Table 3.10: Constraints table

Global constraints (domain related)		
Domain	Constraint	Additional info
Electrical power system	Voltage limits	Violations to be avoided
	Power transfer capabilities	Line capabilities
Control system	Communication performance	Delays, reliability, availability etc.
	Overall control delays	Whole control loop delays
	Algorithm performance	
ICT	Latency requirements	
	Time synchronisation requirements	
Stakeholders	Access issues	Access to data, control etc.

3.3.2 Test System Perspective

In the test case, the objects under investigation have not been isolated, but the test system would require a specific OuI at a time. The two possible OuIs for consideration here are 1) the central controller running the Coordinated voltage control (CVC) algorithm and 2) one real *PV inverter*. Each of the two cases could be evaluated on a similar test system. To formulate the test system, we briefly outline the possible test objectives for either OuI.

Regarding the *CVC algorithm*, the purpose of investigation is to verify the correct operation of the algorithm. The algorithm should be able to detect and mitigate all congestions (voltage or current) during a predefined maximum delay. The test aims also to characterise the performance of the algorithm for instance by comparing the network losses with and without the algorithm. To be able to do that, the test systems has to enable repetition of identical test sequences.

Regarding the *real PV inverter*, the purpose of investigation would be to verify that the inverter implements the set points given to it by the CVC algorithm correctly.

The system under test includes the central controller, the distribution network, all controllable resources controlled by the CVC algorithm (transformer tap changer, inverters of DER units and the storage systems), one of which may be the real PV inverter, and the communication network. The GSC presented in the previous section already included tables for components, attributes and constraints and these tables are not repeated here since they would include almost the same information. The difference is that not all the components, attributes and constraints are relevant for the test case but the graphical representation of the system configuration is adequate to represent the differences. Connectivity is presented in the specific SCs either using connectivity and topology tables or by the graphical representation. In this example case, only the graphical presentation is included.

Further, the test system is more concrete: a specific system configuration with DMS Controller and specific resources and grid configuration. The simple test system includes two loads and three inverters on one spread out feeder. Without further motivation (subject of Section 5), the OuI here is assumed to be the DMS Controller which implements the CVC algorithm.

The same test system is illustrated in Figure 3.11, and in the combination of Figure 3.12 and Figure 3.13.

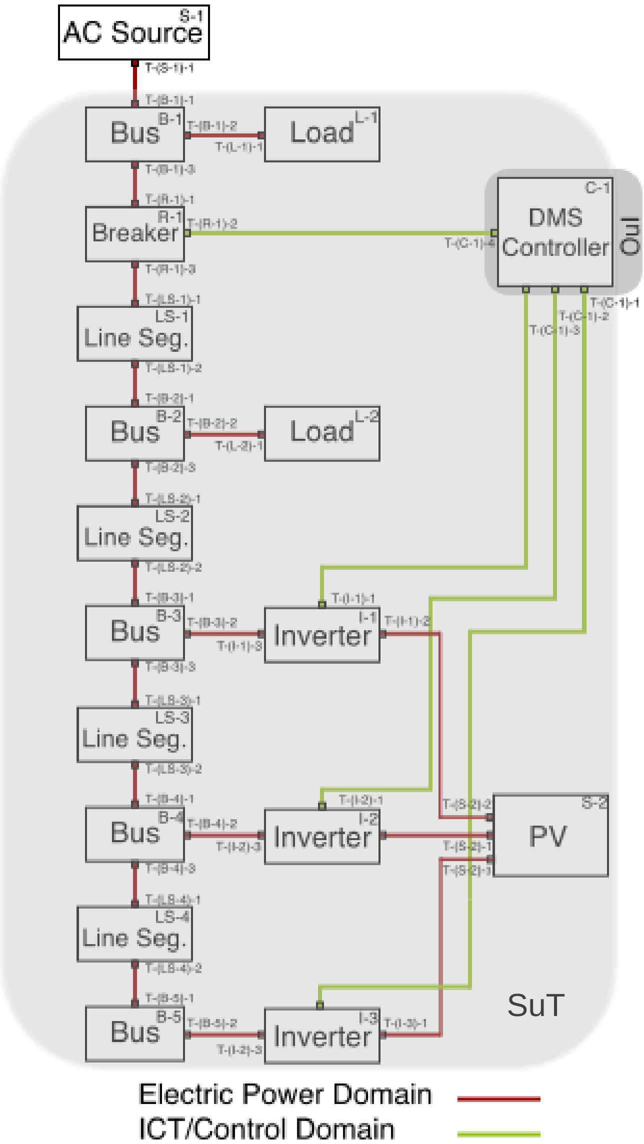


Figure 3.11: Detailed Graphical representation of the TS-SC for the coordinated voltage control test case with OuI as DMS controller

In combination with Figure 3.13, the domain-specific single-line diagram in Figure 3.12 specifies the distribution system, so it is not detailed on the multi-domain system diagram of Figure 3.13. The change from Figure 3.11 to the much simpler multi-domain Figure 3.13 illustrates how a combined approach can be used for simplicity and clarity. Note that annotations of “Distribution System” terminal names in Figure 3.13 T-(B-*i*)-2 *(i=2..5)* correspond to electrical connections at the busses of the one-line diagram in Figure 3.12.

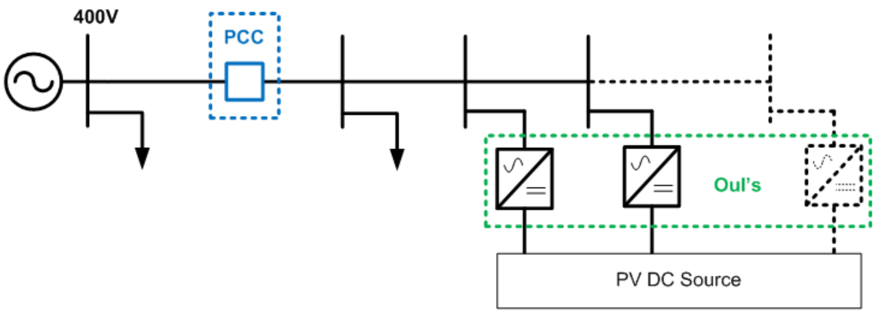


Figure 3.12: One-line diagram (mock-up) of a distribution system domain-specific representation

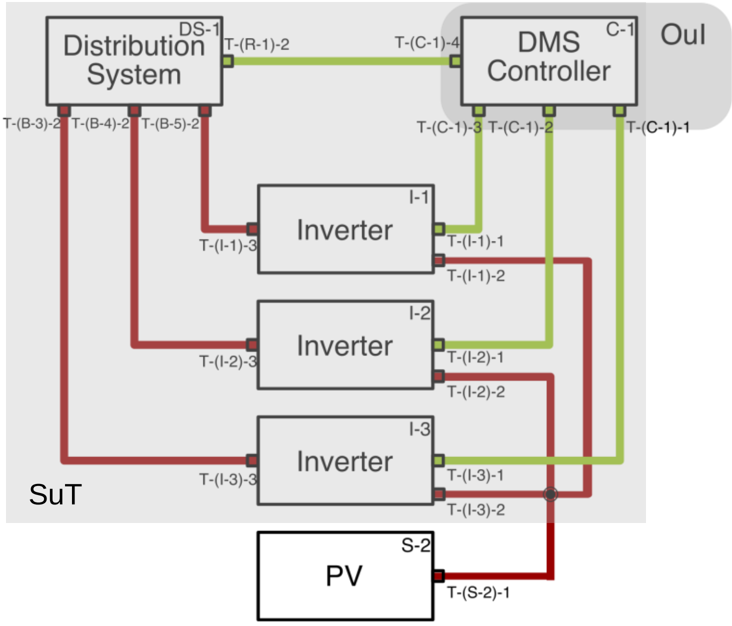


Figure 3.13: In combination with Hybrid graphical representation of the TS-SC for the coordinated voltage control test case with OuI as DMS controller

3.3.3 A Laboratory System Configuration and Mapping of a Test System

In the experiment specification, the test case is mapped to a specific laboratory. The Coordinated Voltage Control algorithm is tested in combined CHIL and PHIL. The combined Control and Power HIL simulation is described such as one of the test network's PV inverters is the hardware under test of the PHIL test and the central controller performing the CVC algorithm is the controller under test of the CHIL test.

The network is simulated in the RTDS (Digital Real-time simulator). The physical PV inverter is connected on its DC side to a PV simulator, in order to allow fully controllable and customizable characteristics for the PV, such as MPP power, irradiance, temperature, etc. In addition, the inverter is connected with a dedicated communication and control interfacing device which enables the transfer of set-points from the controller to the inverter.

In order to connect the PV inverter (power component) to the real-time simulated network, a power amplifier is required. In this setup, a linear 4-quadrant amplifier is used.

The communication of the RTDS with the central controller is implemented with a communication interface, which consists of analog and digital input/output modules and a real-time target (RTT) computer. The I/O modules communicate via EtherCat protocol with the RTT, which in turn interfaces these signals directly with the MATLAB Workspace. Similarly, as in the previous section, also here only the graphical SC representation is given.

The illustration presented in Figure 2.2 on page 14 is thus a counterexample to the recommended approach. Not only can the test system not clearly be distinguished from the lab setup. Also systems and functions are partly mixed. To offer a both concise but similarly intuitive illustration of the overall test configurations, a layered mapping is illustrated in Figure 5.16.

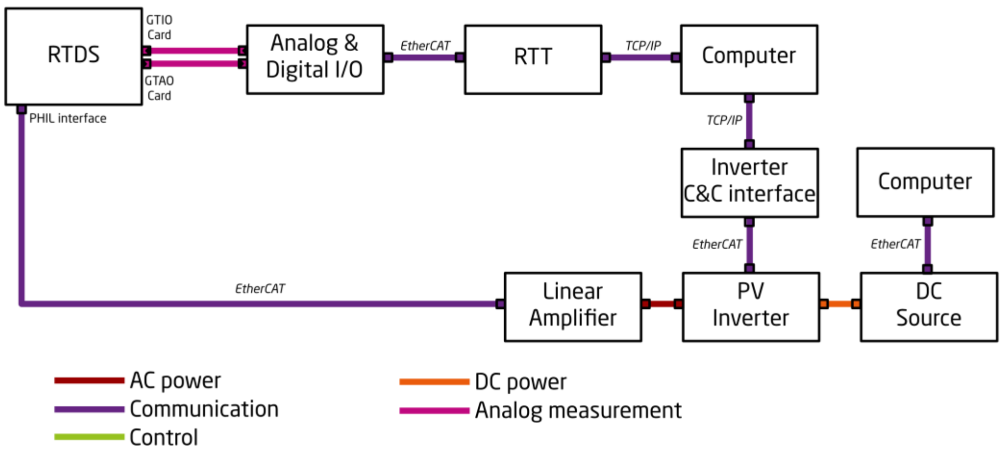


Figure 3.14: Graphical representation of the E-SC for the coordinated voltage control test case

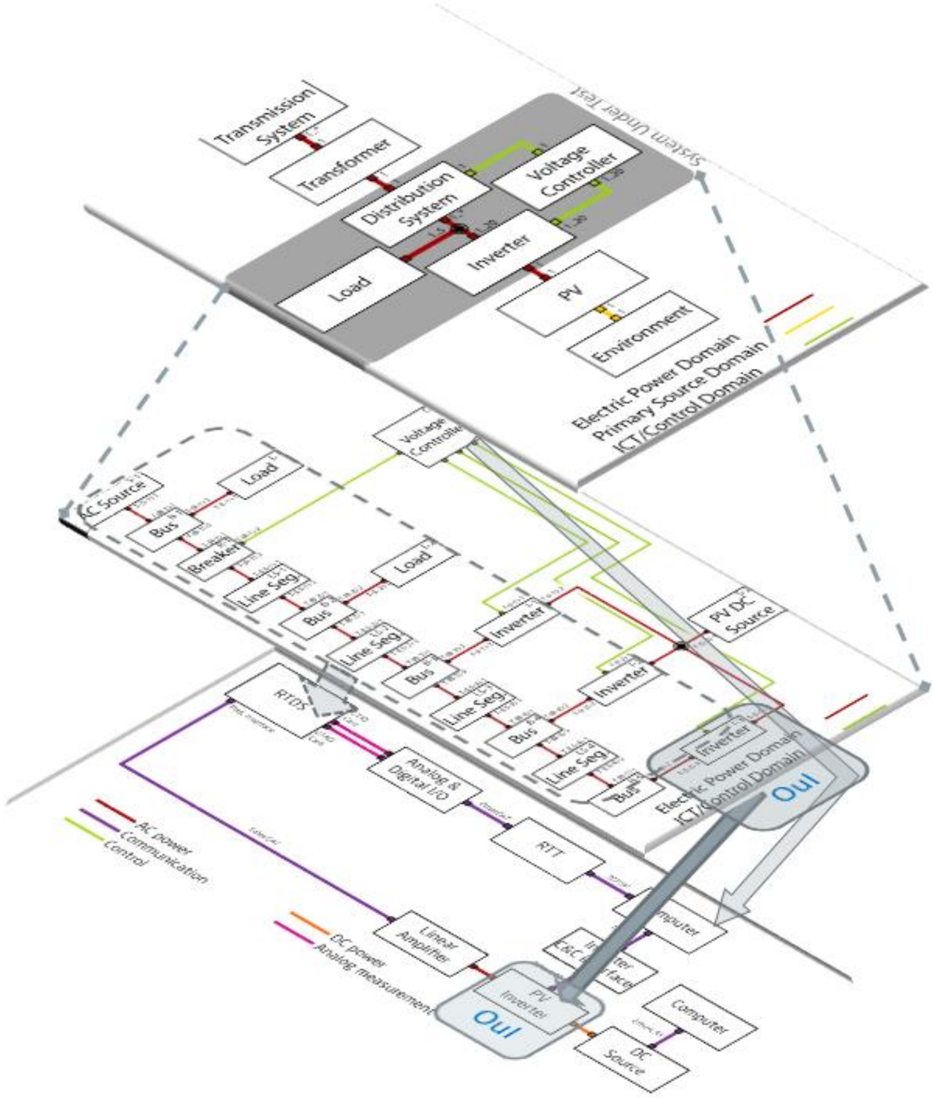


Figure 3.15: Intuitive layering of the TC-GSC, TS-SC and E-SC for the same test description

4 Use Case

The objective of a test scenario typically entails that a given system is required to exhibit certain functions. As reference to formulate concrete properties and thus test requirements for a given function, use cases specify objectives and desired behaviour of a system so that it can be said to exhibit the named function. This summarizes outlines the state of the art on use case descriptions and explains how use cases are interpreted and integrated in the ERIGrid context.

4.1 State of the Art

Use cases are a generally accepted valuable method of documenting requirements for applications and processes for purpose of defining functions of systems interfaces [46]. In the smart grid European context they are considered, by the Smart Grid Coordination Group [47] as a fundamental methodology to be used in order to implement interoperable systems. According to this fact, the concept of use cases has been used in various EU FP7 projects (e.g. Grid4EU [48, p. 7] and ELECTRA [49, p. 7]) to specify functionalities and requirements to be implemented.

4.1.1 IEC Framework

At normative level, the International Electrotechnical Commission (IEC) is developing IEC 62559 standards series within TC8 (“Systems aspects for electrical energy supply”) and System Committee (SyC) “Smart Energy” (which deals with use case methodology). In particular IEC 62559-2 [50] defines the use case template structure (see Figure 4.1) as evolution of the Smart Grid Coordination Group defining a standardized way.

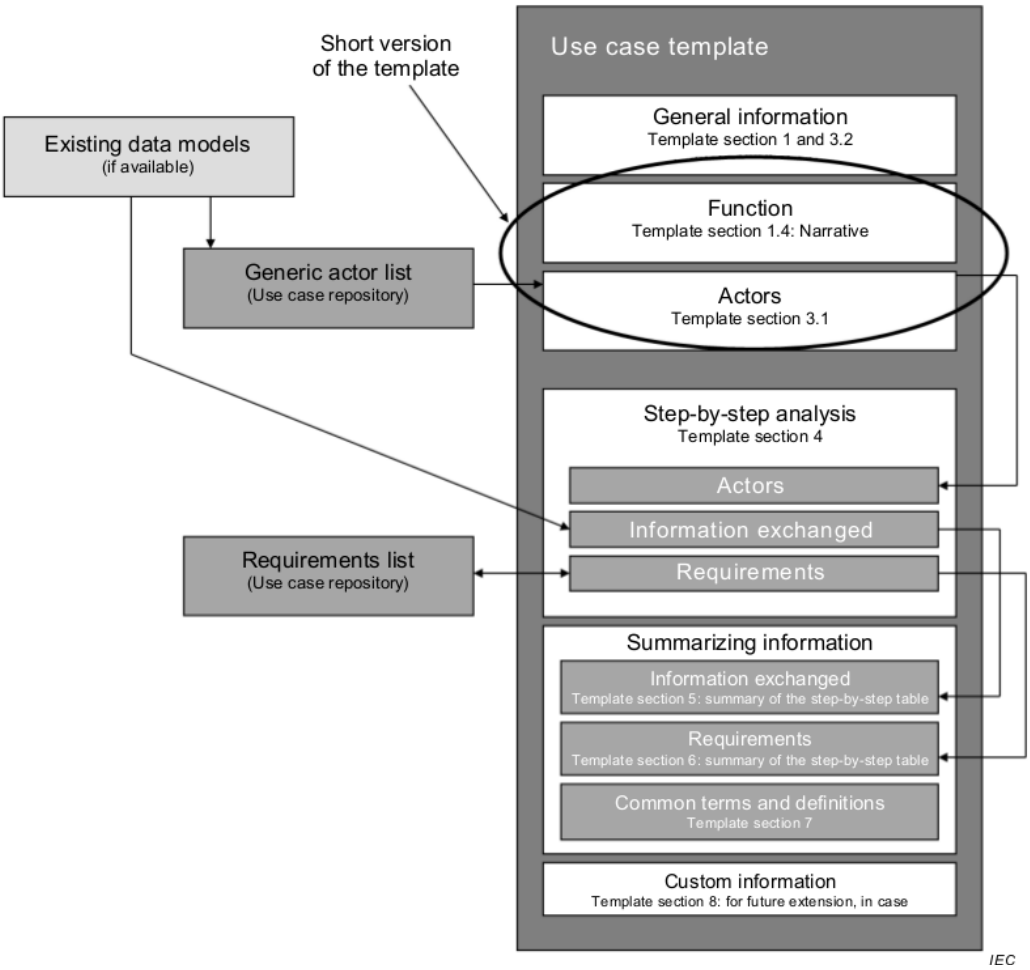


Figure 4.1: IEC 62559 use case template

In the previous figure some parts of the template should be filled with information coming from “Existing data models” and a “Use cases repository”. This last one consists of a software tool that will be made available by IEC and will contain reusable information such as actors and requirements. Existing data models may be used in order to describe messages exchanged in the context of a use case with a standard vocabulary: an example of such data model is the IEC CIM (Common Information Model) [51].

The “Optimal centralized coordinated voltage control” use case example is reported in Annex 9.4.

4.1.2 EPRI Framework

From a historical point of view, the IEC 62559-2 use case template originates from the IntelliGrid program developed by the Electrical Power Research Institute (EPRI). An architecture was defined within this program as a means to implement the “IntelliGrid vision” of the automated, self-healing, and efficient power system of the future.

This template was published inside the standard document IEC PAS 62559:2008, which is now deprecated and replaced by IEC 62559-2; however, there are a good number of use cases available from EPRI which use the old version. As expected, there are many similarities between the two templates, since IEC 62559-2 template is basically an extension of the EPRI one, with some added fields and tables. Examples of information missing from the EPRI template are: version management, key performance indicators, classification information, grouping of actors, and the “scenario” concept. Regarding the latter, the EPRI template has two distinct tables for the “normal sequence” and for “alternative, error management and/or maintenance/backup sequences”, while the IEC 61559-2 template has a unified table that can be replicated for each scenario.

4.1.3 SGAM Methodology

SGAM is a very widespread method that can be used in order to define and implement smart grid systems and functionalities in an interoperable way. It has been developed in the context of the Smart Grid Coordination Group activities and based on the fundamental concepts of domains, zones and interoperability layers (many of these concepts are evolutions of what has been elaborated previously by NIST [52]). Domains and zones are the components of the “smart grid plane”, already illustrated in Section 3.1 of this report.

The SGAM (Smart Grid Architecture Model) consists of five layers representing the objectives and processes, functions, information exchange and model, communication protocols and components. These five layers represent an abstract and condensed version of the interoperability categories introduced in the model of the GridWise Architecture Council (GWAC). Each layer covers the “smart grid plane”, which is generated by electric domains and information management areas (see Section 3.1.1). The model is used to as reference designation for information areas in which the interaction between domains occurs. In order to have a clear presentation and easy handling, interoperability layers GWAC model are aggregated into five abstract interoperability layers (Figure 4.2). However, in case of a detailed analysis of interoperability, abstraction can be unfolded.

Inherently, the use case template referenced above covers relevant specifications to the Information and Protocol layers in the scenario, information exchange and requirements sections.

In order to take into account all different aspects of a given solution, five interoperability layers are defined on top of the smart grid plane as shown in the following figure. Each of these layers identifies a specific interoperability category; hence for the realization of an interoperable function, all categories have to be covered by means of standards or specifications (Figure 4.3).

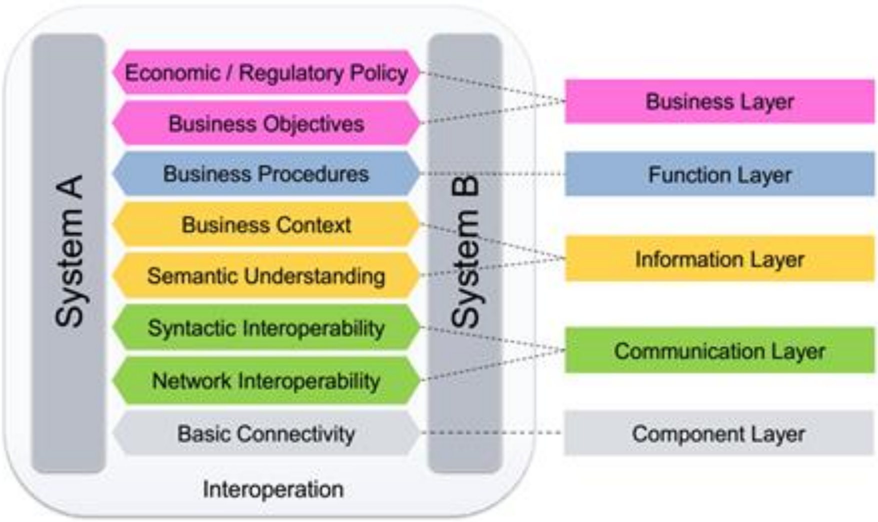


Figure 4.2: Mapping of GWAC-SGAM Layers

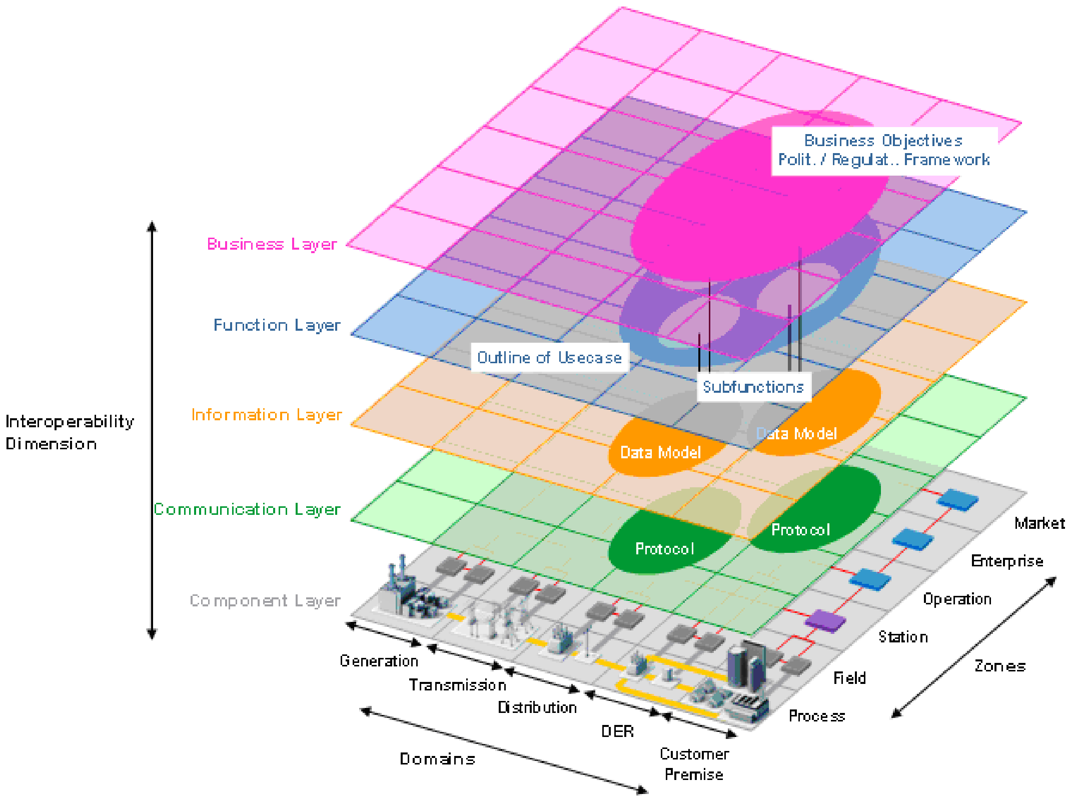


Figure 4.3: SGAM layers and interoperability categories

Functions of the use cases are allocated at the Function layer of the diagram, which is described in [53] as:

> *"The function layer describes functions and services including their relationships from an architectural viewpoint. The functions are represented independent from actors and physical implementations in applications, systems and components. The functions are derived by extracting the use case functionality which is independent from actors."*

To use the SGAM methodology means to take into account all the five layers, thus inserting a use case into a larger analysis of all the possible interoperability issues that should be considered when developing a smart grid solution. The methodology may be applied “top-down” starting from the business layer or “bottom-up” starting from the component layer.

4.1.4 Conclusion on State of the Art

The documentation of the functions provided by the devices and actors of a system has been always considered a fundamental task. In fact, as reported in this analysis of the state of the art, standard procedures aimed at describing the applications that have been proposed since the very first innovations introduced in (electricity) power systems.

The most recent standard for the documentation of use cases is represented by the IEC 62559 for which data models, actors lists and dedicated repositories have been developed during the last years thanks to the adoption of the standard templates (IEC 62559-2) by several research and innovation projects and initiatives in the field of smart grids.

The last version of the standard is comprehensive and includes all the fields required for a detailed and exhaustive description of the use cases. However, the continuous evolution of the power system (particularly speaking of interdependencies between different domains) is a limitation that is expected to appear in ERIGrid. In particular, the standard graphical representations of use cases (such as the SGAM) include only the most common domains and, during the progress of ERIGrid, description methods will be likely subjected to modifications in order to increase their level of detail.

4.2 Approach

The use case methodology is a practice for formal description and specification of functional requirements of systems. A use case is a structured description of the possible sequences of interactions between the system under discussion and its external actors, related to a particular goal. The description is primarily textual, but it can also be based on diagrams, notably the ones defined by the UML standard. In general, the main components of a use case are:

- Name and goal of the use case;
- Actors involved (external subjects interacting with the system in order to achieve the function goal);
- Assumption and preconditions to the Use Case validity;
- Scenario(s) (a sequence of steps in the actors-system interactions);
- Trigger events that start different possible scenarios.

In this context, an actor is every entity having behaviour and interacting with the system under discussion to achieve a specific goal. Actors can be for example humans, pieces of hardware or even other software systems. An actor list example is reported in appendix (“Actor list example”).

In the above description, the system under discussion is treated as a “black box” and its internal structure is not described in the use case. Sometimes, however, it can be useful to show interactions between its different sub-systems, and in this case a “white box” description is obtained.

A first overview of a use case may be given graphically by a UML context diagram, which shows the system operational context with external actors that interact with the system for the achievement of the use case goal (Figure 4.4). In the context of ERIGrid, the “System” implementing the use case is the “system under test” of a test case, and in particular this system will be contained in a system configuration as defined by ERIGrid.

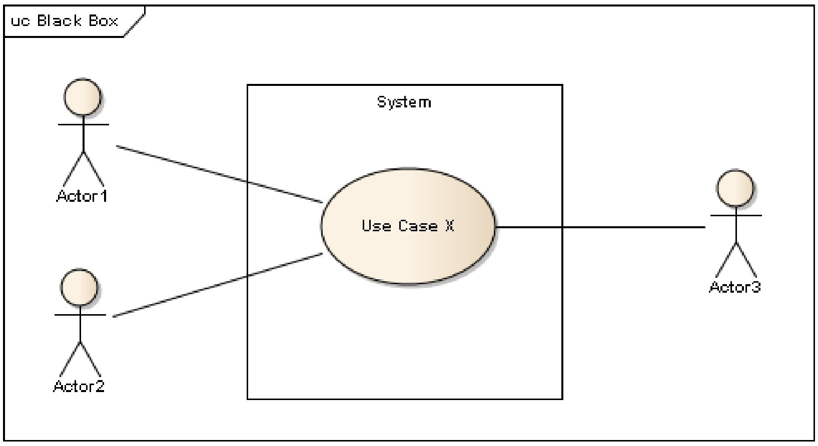


Figure 4.4: UML context diagram example

The context diagram is thus simple, but on the other hand gives an immediate overview of the actors involved in the use case and of the system implementing the use case. To describe the actual exchange of information between the actors, a scenario should be defined: it may consist in a sequence of interactions between the subject system and the external actors. It defines one of several possible paths in the description of sequences (e.g. all the foreseen interaction steps succeed and the goal of the function is obtained). In this case scenarios may be depicted using a sequence diagram (Figure 4.5).

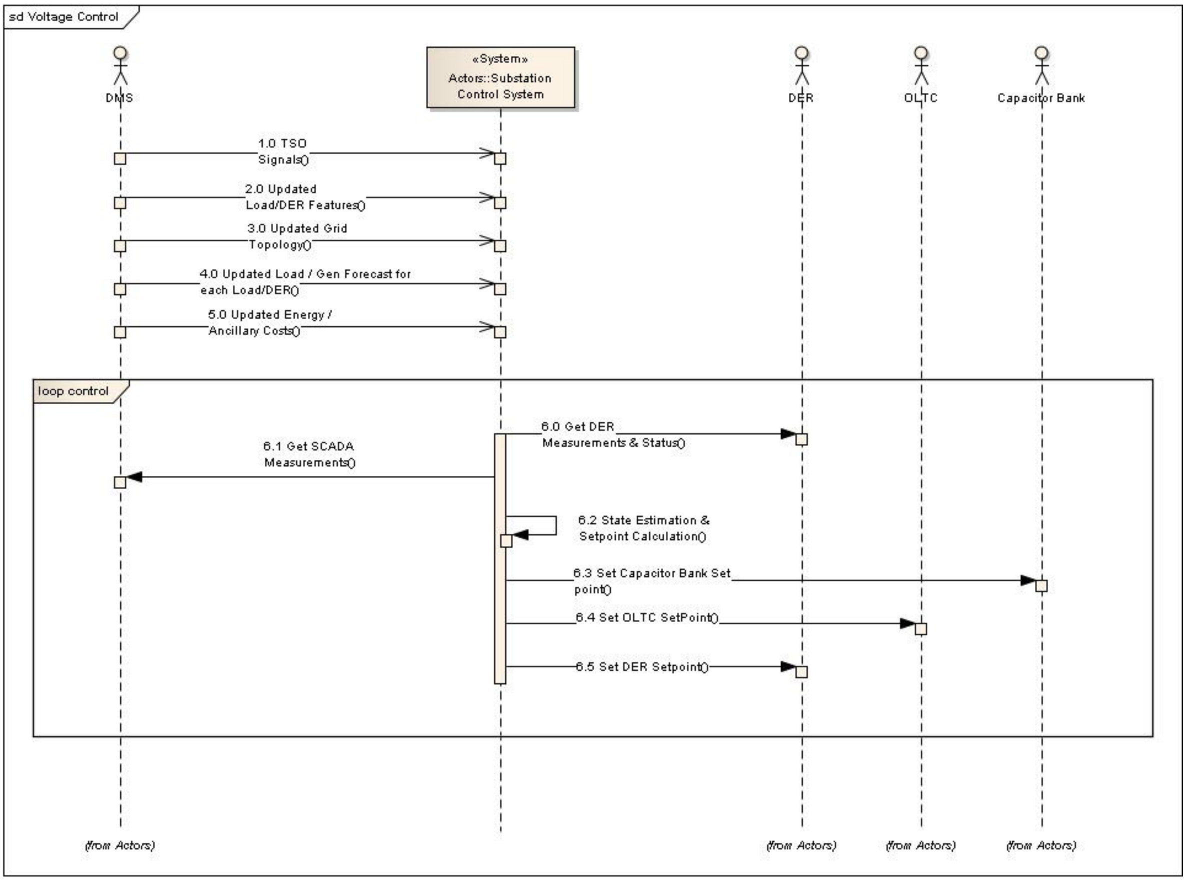


Figure 4.5: Sequence diagram example

An example of the application of the IEC 62559 template relevant to the example scenarios in this report is included in Annex 9.4.

4.3 Integration in the ERIGrid Holistic Testing Framework

The chosen use case description methodology is of general applicability for the functional description of smart grid systems. For the specific context of ERIGrid, it is worthwhile to describe how this methodology can be integrated into the broader process of "holistic testing" described in Section 2.3 of this document. Specifically, a connection between single use cases and related system configurations should be possible. Also, a mapping between test cases and use cases must be considered in order to handle the whole process.

The connection between use cases and system configuration has been considered also in ERIGrid D-JRA1.2 for the more specific task of selecting the project's focal use cases: the starting point is to consider the available system configurations (Distribution Grid, Transmission Grid and Offshore Wind, Vertical Integration) [54].

The connection with use cases is conveyed by the components: each system configuration provides services by means of its components (the description of components for each system configuration can be found in Deliverable D-JRA1.1). The services that are relevant to ERIGrid, as described in D-JRA1.2, are the following:

- Energy balance;
- Energy efficiency;
- Power quality;
- Power system stability;
- Infrastructure integrity, protection and restoration.

In this context, a use case is a function used to provide one of the above services to a given system configuration. The following image provides a schematic illustration of the links between use cases, system configurations and system services in the context of ERIGrid. In Figure 4.6, the system configuration contains different systems (directly related to its components): the use case is implemented by one of these systems (see also Figure 4.4).

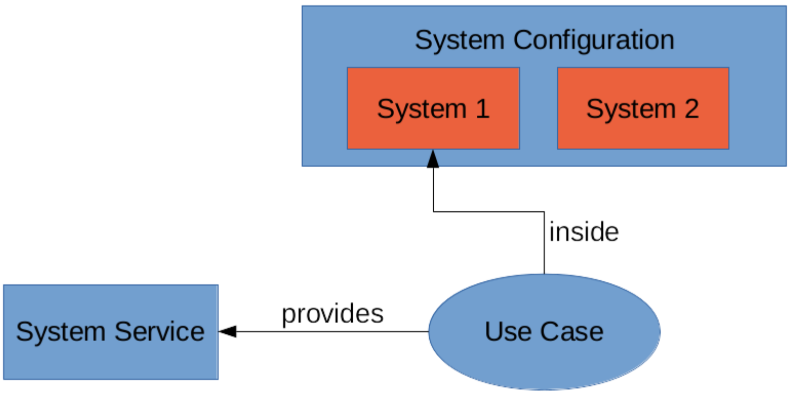


Figure 4.6: Relationships between use cases, system configurations and system services

A system that fulfils a given use case is said to implement the function stipulated by a use case. This direct dependency between use cases and functions can lead to confusion as to whether a use case *is* a function, is dependent on a function, or whether a function implements a use case. This relationship is intentionally left open – a system architecture will define the composition of functions: whether a use case is fulfilled directly by a single system function, by a combination of system functions, or by a certain behaviour of a given function should be left to the system development. However, as the use case expresses the required system behaviour in a given context, the functional and non-functional requirements to be validated or verified in a test case, can be derived from the relevant use cases.

5 Test Case

As introduced in Section 2.3, the steps of the holistic testing procedure comprise first, a *holistic test case*, a subsequent mapping, resulting in a *test specification* of one or more sub-tests, which then are mapped to concrete research infrastructures (RI) in respective *experiment specifications*. The holistic test case refines system configuration, use cases and test objectives. Since the holistic test is a generic description, that may imply several parallel or sequenced sub-tests, these sub-tests need to be detailed with more specific tests systems. The test systems for sub-tests must be specified such that it is possible to conduct them in a single experiment setup. That experiment setup is a realization of the test system at a specific and suitable RI. This section motivates and describes in sequence all three description aspects - the holistic test case, the sub-test specification and the experiments specification.

This section deals with identifying requirements and relevant information needed to define the respective description methods in terms of the ERIGrid holistic testing approach. In the holistic approach, RIs capable of performing the required tests must be found. In order to choose RIs - and thus prepare the mapping steps - some higher level of information about the tests that are incorporated in the (holistic) test specification are needed. After having chosen suitable RIs it is possible to formulate concrete experiment specifications based on the given lab infrastructure.

In Section 5.1, we present a summary of the current practices at the partners' RIs (Note that the complete analysis of these practices is reported in Appendix 9.5; this work was developed as an input for the development of the test description methodology), and review several state of the art practices. Following that, the holistic test case, test specification and experiment specification approaches are presented along with exemplary cases in Sections 5.2, 0 and 5.4, respectively.

5.1 Current Practices and State of the Art of Test Specification

This section summarises current practices of the ERIGrid partners' RIs in Section 5.1.1. An overview of metamodelling is presented in Section 5.1.2. Furthermore, examples of test cases and test case description for both physical and software testing are given in Sections 5.1.3 and 5.1.4, respectively. A standardised approach for domain independent testing is presented in Section 5.1.4.3. Section 5.1.5 concludes with a discussion of the presented approaches.

5.1.1 Current practices at ERIGrid Research Infrastructures

In order to get an overview about the current practices regarding testing procedures and test cases at the ERIGrid partners' research infrastructures (RI) a questionnaire (see Appendix 9.5.1) has been filled, covering certain aspects such as type of test, purpose of test, or input-output relations. In total 28 testing procedures partly describing several slightly different test cases at once were answered by the ERIGrid consortium.

Aiming at identifying characteristic properties required to unambiguously specify test cases, five working groups have been formed; each trying to find clusters of similar test cases based on one of the categories of the questionnaire: *object of investigation*, *purpose of investigation*, *test criteria*, *test design*, and *test setup*. The findings have influenced the definition of holistic test cases presented in Section 5.2.

Furthermore, it has been investigated which connections there are between different domains or within domains to determine the gaps that appear when aiming for coupled tests. Another working group investigated how the design of experiments methodology could be utilized in order to generate models for exchange based on test results for "offline" representation of tests.

The findings of the five working groups are summarized below.

Object of investigation

Based upon the questionnaire answers categories for the objects of investigation that are tested at the different RIs were created. An overview of the categories and their distribution can be seen in Figure 5.1, where it can be seen that algorithms and power electronic devices are the two most common objects investigated, while there is an even distribution over the other categories.

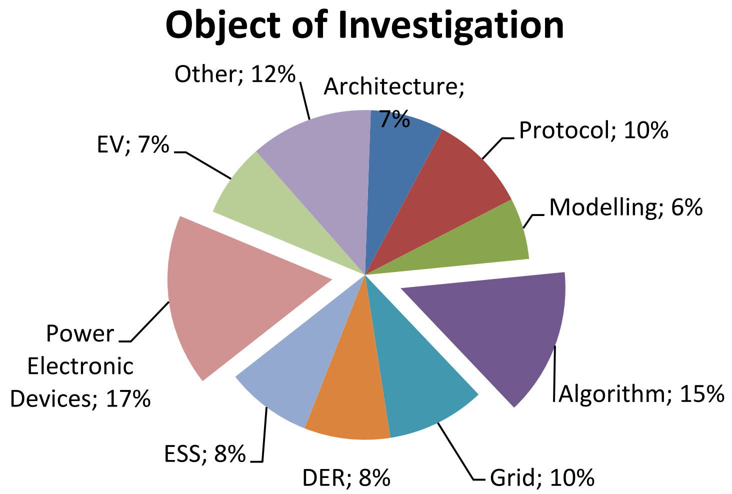


Figure 5.1: Categories and distribution of the Object of Investigation tested at the RIs

Purpose of investigation

Seven categories of purpose of investigation were identified:

- Verification of service provision, functional behaviour and conflict analysis in system-integrated approach
- Performance evaluation of algorithm or equipment
- (Sub)Scenario assessment and validation
- Performance and response time of protection equipment
- Compatibility and interoperability of ICT components
- Cyber Security of ICT

Test criteria

The test criteria have been separated in two types according to their properties: characterization criteria and validation/testing criteria. Within characterization criteria, two types of criteria are defined, those related to behaviour and those related to performance. For validation/testing criteria, there are those related to undesired behaviour, for testing against one or more characteristics, in concordance to standards, validation of accuracy/functionality, and qualitative criteria.

Also, within this analysis, a distinction between single domain tests and multi-domain tests was found. Single domain tests are those where the object of investigation is within the electrical domain, another physical domain (e.g. heat), or ICT domain. The multi-domain tests are those where the interaction between these domains is taken into account, e.g. for energy conversion.

Test design

The test design is affected by the purpose of the investigation and the test criteria. In Appendix 9.5.5 a thorough overview is given over the different test designs used in the consortium, but here it suffices to point out that a generic test flow was identified (see Figure 5.2).

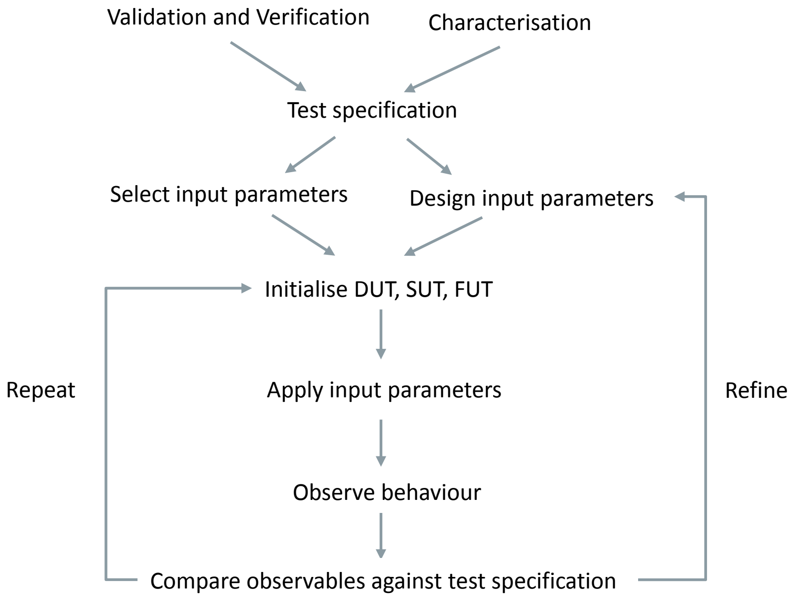


Figure 5.2: Generic test flow:
DUT- Device under test, SUT – System under test, FUT – Function under test

Test setup

Eleven clusters with respect to test setup were identified:

- Hardware Set up
- Software Set up
- In field Test
- MV test capability
- Hardware and software integration
- Real Time and HIL
- HW Grid and components simulator/emulator
- Real scale components
- Interoperability and communication
- Automated test
- Co-simulation or multi-domain simulation test

As can be seen from the clusters, the RIs in the consortium cover a wide spectrum of test setups, which also reflects the complexity of test cyber-physical energy systems.

5.1.2 Metamodelling / Surrogate modelling

In Section 2.2.2 the concept of Design of Experiments (DoE) was introduced as a means for systematic testing and evaluation: After systematically choosing combinations of input parameter values, an empirical model of an outcome can be fitted to the data - a so-called metamodel or surrogate model. This could, for example, be a model that predicts the time needed until an algorithm terminates depending on the inputs or the amount of computing resources it would require. The general idea of DoE is to optimize the selection of experiments needed to be executed in order to create such a model. The objective of this optimization can be based on the cost, simplicity and / or effectiveness of experiments. The concept of metamodeling employs the same DoE methodology, but it expects the system responses that are later predicted by the metamodel to be the outputs of the system. Thus, if the metamodel is deemed good enough, it may be used as a replacement for the original model. These simplified black-box models are usually less accurate but may be executed much faster than the original experimental setup and it is possible to reuse them outside the original setup.

Domains of application

According to a brief literature research to obtain an overview, surrogate modelling is heavily used in hydrological, aerodynamic, electrical systems, vehicle modelling and others. The latter contains all experiments that did not fit well with a domain or the domain was not clear.

Size of modelled systems / number of factors and responses

In most experiments that have been mentioned in the chosen publications, up to ten factors, i.e. influenceable input parameters were considered. Systems with up to 50 factors are rarer but still feasible. The maximum number of factors considered in a single experiment was 632. Up to twenty experiments were listed without giving an exact number of factors. Most of the other experiments have a single response. Two and Three responses are also quite common. Higher number of responses up to 14 are rare but still feasible.

Surrogate Simulation Models

Current work in progress at OFFIS proposes to utilize the metamodeling process to replace simulation models in co-simulation setups with semi-automatically generated surrogate simulation models.

As depicted in Figure 5.3, such a surrogate simulation model embeds a metamodel of an original model and adds mechanisms to store and update the current simulator state to it.

The approach starts off with a detailed description of the original models' characteristics (categorization of parameters, parameter names and types, hidden inner states) and the specification of test simulations, which is then used to generate a sampling plan. The responses of the model given these input samples are then observed in a series of simulation experiments in order to create an input-output dataset. Regression models are then fitted to the dataset to predict / approximate / interpolate the outputs of the original model.

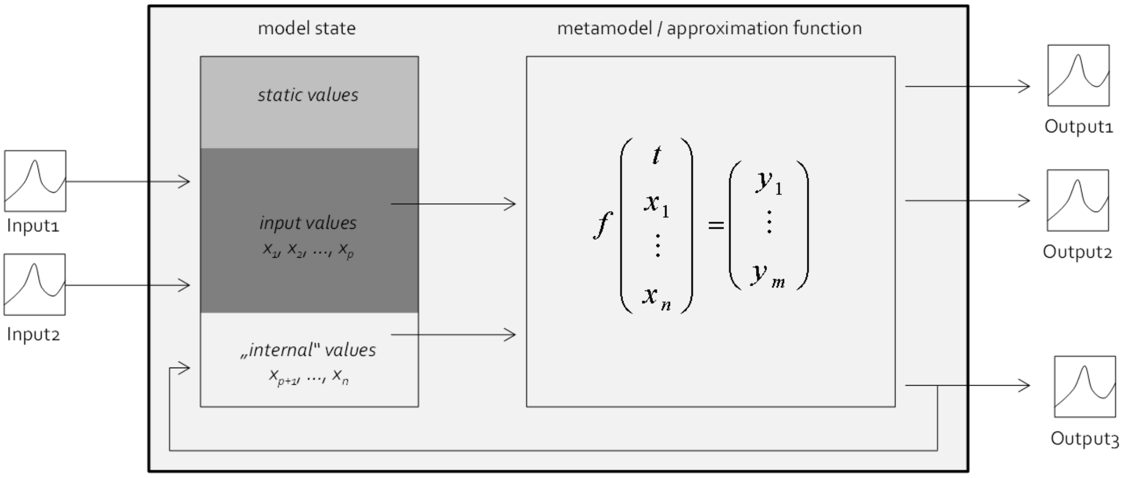


Figure 5.3: Surrogate Simulation Model

The following three phases in the targeted process were identified:

- *Preparation:* During this phase the experimentalist gathers all required information and prepares the experimental setup.
- *DoE (Metamodeling):* In this phase a model is fitted to the input-response dataset.
- *Exchange:* During the exchange phase (or simulation phase) the metamodel has been distributed to another RI and can there be used a co-simulation scenario.

The partners were asked to gather all requirements and challenges they could think of regarding the preparation and exchange phases as well as general in challenges and limitations. In the preparation step, the focus was on the identification of input and output parameters. Input parameters were mostly distinguished into controllable / adjustable and non-adjustable / constant input parameters. Sometimes they were further characterized as “relevant” or as having considerable influence on the outcome of the experiment.

The tasks necessary to exchange surrogate models in ERIGrid are not yet clear. Additional information about the surrogate model should be provided before it can be passed to another RI, such as:

- The type of simulation or use case that the model is suitable for use in,
- All the choices of parameter levels and ranges,
- Information about the quality / approximation error of the surrogate model,
- A description of the interface.

It was also mentioned that the surrogate model should be embedded in a Functional Mockup Unit (FMU) for either model exchange or co-simulation.

The contributions stated limitations of three different types:

1. *Methodical limitations*
   - Bad sampling strategy: Some regions of the sampling space might be covered worse than others, resulting in rougher approximation for some inputs than others or certain behaviours simply not showing because the corresponding samples are not represented.
   - List of possible input parameters is too large and unmanageable.
2. *Limitations regarding the capabilities of surrogate models in general*
   - A single surrogate model cannot approximate all possible behaviours and their variations. Therefore, depending on requirements a large amount of surrogates might be needed. It always generates extra effort to create an additional surrogate.
   - The surrogate might be too slow in order to use it in real-time experiments
   - It might not be possible to represent discrete events when simulating with surrogate models
3. *Technical limitations*
   - The simulation tool in use does not support an integration into the metamodelling process
   - The computational effort in order to train a model is large

5.1.3 State of the Art on the Physical Testing in Power Systems

Standardized testing in context of power systems involves testing of a component e.g. an inverter of a distributed generator in an open-loop environment, in which commonly a grid simulator applies predefined voltage and frequency profiles.

However, in this way, possible interactions with other components e.g. inverters, On Load Tap Changers, synchronous generators etc. and the system as a whole are neglected. As the complexity of electricity networks and components is constantly increasing, conventional component testing is proving to be insufficient [55]. According to DERLab’s “European White Book on Real-Time Power Hardware-in-the-loop testing” [56], for the purpose of de-risking equipment in complex grids under dynamic conditions, testing should include the entire system rendering the combination of simulation and hardware experimentation inevitable.

From a system perspective, typically offline digital simulations are performed to investigate power system phenomena using mathematical models. However, the large-scale deployment of complex devices such as power electronics based DER with advanced functionalities poses new challenges for simulation, as these devices are particularly difficult to model accurately and might be involved in complex interactions within the power system. Therefore, Power Hardware-in-the-Loop (PHIL) testing can reveal system related phenomena that are not visible in pure digital simulations.

We present two examples of test cases corresponding to the state of the art. An exemplary test case specification is given as well.

5.1.3.1 Component Testing

Component testing can be seen as the equivalent for unit testing in the context of software development (see Section 5.1.4). According to [11] “a unit could be considered as a single device.” The focus lies on the individual component. Component testing is typically performed as an open-loop test, where predefined profiles e.g. voltage, current or frequency are applied to the Device under Test. The device under test does not affect the test conditions.

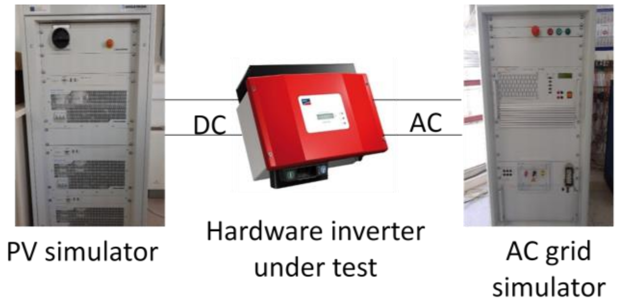


Figure 5.4: Example of component testing – PV inverter test setup

An example of component testing is the compliance testing (according to standards) of a hardware photovoltaic inverter. A typical setup can be seen in Figure 5.4. The PV inverter is connected to a PV simulator and an AC grid simulator. The PV simulator is a DC source, which can simulate a PV string (I-V characteristic curve) and controllable environmental conditions (solar radiation, temperature) and it is used to perform tests on the DC side, such as Maximum Power Point Tracking (MPPT). The AC grid simulator can produce user-defined voltage and frequency profiles and gives the possibility to perform tests on the AC side of the inverter, e.g Fault Ride Through (FRT), Q(U) droop control test, frequency tests etc.

In order to illustrate how the conventional component testing approach is adapted to new technologies utilizing ICT, the following example is extracted from [57].

With the introduction of numerical digital relays and more sophisticated microprocessor-based test sets, open and closed loop (real-time) digital simulators and open loop playback equipment have been introduced. Although the protection functions of digital and analogue relays are essentially very similar, the digital technology offers new possibilities and new problems due to the increasing complexity of relays and the trend towards communication based scheme logics as opposed to more traditional hardwired approaches.

Due to the increasing complexity and the great importance of some applications, type testing has become increasingly more important. The following describes different types of type testing, including both static and dynamic approaches.

5.1.3.2 Classification of Type Testing

Type testing can be classified as certification or application conformance testing. Certification type testing is normally performed by a certification organization or by a testing company under the supervision of a certification organization, while application tests are performed by a manufacturer or a testing company on request for a specific end-user, for example a utility [58]. The way certification type tests are performed varies in each country, depending on the existing regulations. Certification type tests are normally performed:

- At the end of development of a new protection relay
- After software upgrade of the protection relay
- After the addition of a new function
- After hardware upgrade of the protection relay

Certification type tests concern normalized tests under normalised procedures, the so-called conformance and performance tests, which aim at verifying the conformance of the protection relay against its specifications. These tests are generally related to international standards, such as IEC 60255 and ANSI C37.90.

However, compliance may also involve consideration of the requirements of IEC 61000, 60068 and 60529, while products intended for use in the EEC also have to comply with the requirements of Directives 89/336/EEC and 73/23/EEC [59]. Since type testing of a digital or numerical relay involves testing of software as well as hardware, the type testing process is very complicated and more involved than a static or electromechanical relay.

Functional conformance tests verify the functionality of the protection against the test standard specification. The functional conformance tests consist of applying the appropriate inputs to the relay under test and measuring the performance to determine if it meets the specification. They are usually carried out under controlled environmental conditions. The testing may be extensive, therefore to minimize the required time to perform these tests, dedicated test sets have been developed [60]. These types of tests are also called static type testing.

Application tests are carried out to demonstrate that a protection scheme is capable to protect a type of network under certain fault conditions. These tests are normally requestedby the end-user, such as a utility:

- To study the behaviour of protection relays in a particular power network before new protection installation or change in the primary system
- In case of troubleshooting mal-operation
- To optimise settings in case of complicated networks

Application tests are based on the use of transients for testing protective relays in order to simulate the dynamic behaviour of the network during faults and test the protection system performance.

Generally, there are two ways of creating type tests:

- Using transients obtained from recorded or calculated waveforms. In this case, a test set can be used to playback a recorded waveform or to generate a calculated waveform that represents a transient of the network. (static type testing)
- Using transients calculated in real-time. In this case, a real-time digital simulator is used to simulate the network during normal and fault conditions. (dynamic type testing)

Static type testing

Static type testing consists of applying inputs to a protection relay and measuring the performance to determine if it meets the specification or not.

Static type testing is normally extensive and it includes a high number of tests. For example, considering an overcurrent protection relay, some of the typical static type tests are: three phase pick-up and drop-off accuracy, accuracy of Definite Time (DT) timer, accuracy of Inverse Definite Minimum Time (IDMT) curves, accuracy of reset timers, etc. All the tests are done over the complete range of settings.

One example of static type test technique is the ramping technique, which is used to determine limiting values, such as minimum pick-up or switching hysteresis (e.g. pick-up/drop-off ratio). The software controls the amplifier and commands it to generate ramps of amplitude, phase, or frequency for the current and voltage outputs and the response of the relay is recorded automatically. This is normally done using a protection test set which generates series of waveforms and records the response of the protection device. Figure 5.5 shows the typical static type testing hardware environment [57].

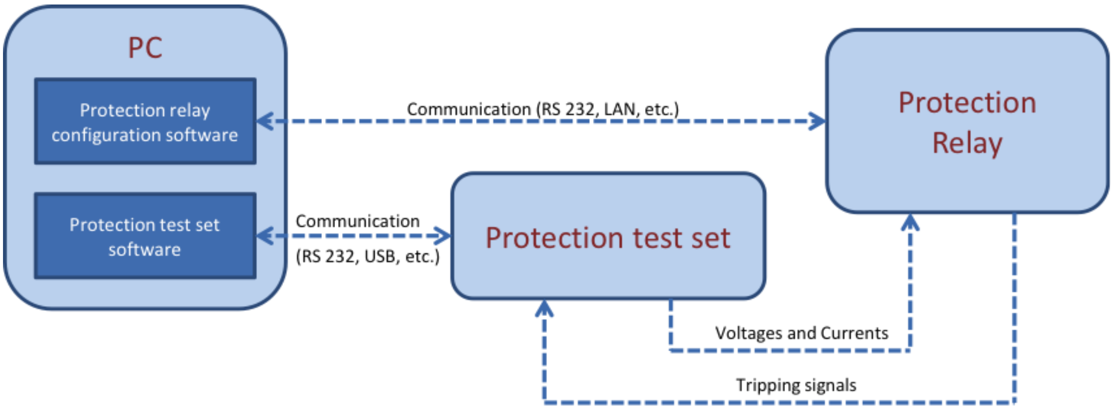


Figure 5.5: Static type testing hardware environment

The hardware normally involved in a static type testing is: a protection test set, a personal computer and the protection relay to be tested. The personal computer is used for both configuring the protection relay using the relay's manufacturer software configuration and to manage the protection testing using the protection test set software. The protection test set generates voltages and currents and records the tripping signals received by the protection relay. If the protection relay is fully IEC 61850 compliant, sample values substitute the analogue values and GOOSE messages are used instead of signalling via relay contacts.

Dynamic type testing

Dynamic type tests consist of simulating transients of a network model in real-time to dynamically demonstrate the satisfactory performance of protection relays.

In the past, dynamic type tests were conducted by using physical scaled down models of electrical power systems, e.g. artificial transmission lines. However, these models had significant limitations in the current and voltage waveforms that could be generated and their use required a lot of time because testing automation was not possible.

With the evolution of microprocessors, a new generation of real-time digital simulators based on distributed microprocessor hardware has been developed and it is now widely used to conduct closed-loop testing of physical devices, including protection relays and protection schemes.

Real-time power system simulators are a combination of advanced computer hardware and comprehensive software. These simulators can solve the power system equations fast enough to continuously produce output conditions that realistically represent conditions in the real network. Therefore, the physical protection equipment can be connected in a closed-loop regime with the power system model and can be subjected to virtually all possible faults and operating conditions including complex fault scenarios with multiple relays and communication channels [61].

Figure 5.6 shows the typical dynamic type testing hardware environment, where the real-time digital simulator simulates the primary system and the protection scheme to be tested, which can be formed by one or more protection relays, receives voltages and currents amplified by slave analogue amplifiers.

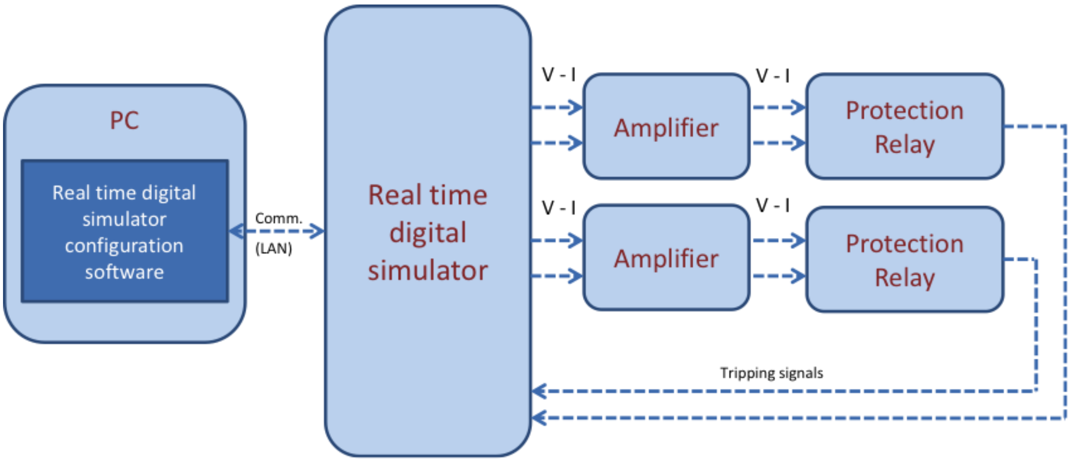


Figure 5.6: Dynamic type testing hardware environment

When the protection equipment is IEC 61850-9-2 sampled value compliant, voltages and currents are provided to the protection equipment through a dedicated interface able to send sampled values to the protection equipment instead of using the amplifiers.

The protection response to faults, such as trip and reclose signals, is then sent back to the simulator to operate the breakers modelled in the simulation. If the protection provides signals via conventional dry contacts, the signal will be received by the simulator using its digital input card, while if the protection equipment is IEC 61850 compliant the breaker commands can be imported into the simulation using a dedicated IEC 61850 interface card.

With the real-time simulation and the protection equipment connected in a closed-loop regime the protection can be subjected to a myriad of faults and operating scenarios. The faults and operating scenarios can be run manually or using automated batch files. The automated batch is often applied to protection system testing where faults are repeated again and again with small changes to the fault inception angle, fault type, fault location, etc. In this way the overall time of testing is significantly reduced.

The previous illustrative example showcases the complexity of testing modern technologies. Yet, it is explained in [55] that the analysis, testing and validation of power system phenomena or components, is most appropriately performed by using the *full hardware system* in controllable, flexible and repeatable conditions. As this is rarely feasible, offline digital simulations are typically performed to investigate power system phenomena using mathematical models [62], and hardware experimental set-ups are built for small systems or component testing. The large scale deployment of power electronics-based distributed generators with advanced functionalities poses new challenges for simulation, as these devices are particularly difficult to model accurately and might be involved in complex interactions within the power system. It should be noted that according to standards, conventional component testing is performed in an open-loop environment, in which commonly a grid simulator applies predefined voltage and frequency profiles to the device without taking into account interactions with other components and the whole system [63]. Thus, for the purpose of safely and thoroughly testing equipment in complex grids under dynamic conditions, it is suggested [56] that testing includes the entire system rendering the combination of simulation and hardware experimentation inevitable.

5.1.3.3 Power System Testing

The component testing approach faces important limitations as it examines only the operation of a specific device; therefore interactions of several components within the power system are neglected [55][63]. As power systems are becoming more active and complex, component testing can no longer be considered sufficient [56] and there is a clear need for more advanced testing methods. In general, system testing considers one individual device integrated in the system [11]. Power system testing is a system level testing method applied on a single domain (i.e. electric power). An example is illustrated in Figure 5.7, where a PHIL test is considered. Equally a full hardware or pure software setup of the shown system can be considered.

It is worth mentioning that the same investigation could be performed in pure simulation, however as the PHIL approach allows the interaction between real hardware and a simulated system, it offers better insight on the actual interactions, i.e. the oscillations not visible in pure simulation**.**

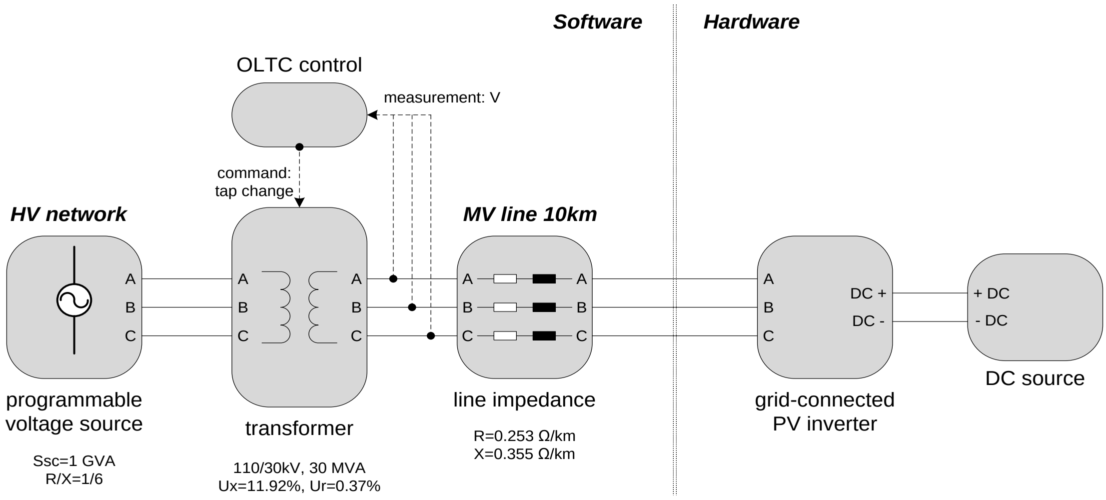


Figure 5.7: Example of Power System testing – Interactions of OLTC and PV inverter (single domain)

5.1.3.4 Test Case Specification

This section gives an overview of how a standard test specification is developed and presented at the Power Networks Demonstration Centre[9] to conduct system and component tests. The main elements of the test specification are described. The format developed for the test specification provides both a brief summary of the test objectives and what is being tested as well as a reference to the main test criteria and how they can be assessed. The test specification is normally accompanied by supporting documentation to elaborate specific information related to the configuration of the test environment and operating procedures for specialised pieces of equipment.

1. *Background and aim of testing*: This section of the test specification describes the background of the system under test (SuT) or device under test (DuT) in the context of the application by the end user. Furthermore, an outline of the test specification and overall aim of the testing is provided.
2. *Description of SuT/DuT:* A description of the main functionalities and performance attributes of the SuT/DuT that are relevant to the test are outlined in this section. Pertinent flow or sequence diagrams are also included here to illustrate the functionality.
3. *Description and configuration of test environment:* This section provides the following information about the test environment in order to achieve the test objectives:
   - Medium voltage (MV) and/or low voltage (LV) test network topology.
   - MV and/or LV equipment used and their configuration (e.g. switchgear, transformers, IEDs, power supplies, etc.).
   - Physical or virtual connectivity of the SuT/DuT in relation the test environment.
   - Configuration of SCADA or ICT systems used for the purpose of the test.
   - Description of simulated elements such as RTDS based tests.

[9] http://www.strath.ac.uk/pndc/

4. *Instrumentation and measurements specification:* A description of the measured attributes and, where appropriate, associated uncertainties are defined in this section. The following information is also included:
   - Sampling rates and bandwidth.
   - Format of stored measurement data.
   - Specification of direct, derived, real-time and post-test processed measurements.
   - Location of physical measurements in relation to the SuT/DuT.
   - Specification for additional instrumentation if the instrumentation installed in the test network does not meet requirements.
5. *Testing schedule:* The defined tests are then summarised in a tabular format (as in Table 5.1).
6. *Health and safety considerations:* Tests carried out using the HV or LV network and equipment require a statement of the risk control measures to ensure that exposure to hazardous situations is eliminated, minimised or controlled as appropriate. This can have a bearing on the measurements, test sequence and test environment configuration.

Table 5.1: Testing Schedule Example

Test ID	Test description
A unique identifier for each test is included here	Objective: the objective for each test is given here. For instance, the objective of a test may be to verify the operating time of equipment performing protection functions. Initial conditions: the specific configuration of the test environment as well as values of attributes prior to conducting a test are defined here (e.g. pre-test loading conditions, energy storage state of charge, bus voltage, etc.). Test sequence: this defines the physical or simulated sequence of events and/or input signals used to initiate the test or stimulate the SuT/DuT over a pre-defined period of time. Assessment criteria: a quantification and/or qualification of test success criteria are defined here. This is normally based on the measured attributed observed against the test sequence in the context of the test objective or expected behaviour of the SuT/DuT.

5.1.4 State of the Art on Testing in Software Development

Apart from tests on physical components or systems, testing in the software development process is as important. A brief introduction is given in this section

5.1.4.1 Software Tests

The following section is based on [64]–[66]. As in the development of other products, testing is an important task in software-development processes. The purpose of software tests is to ensure that a software system works as expected when used by the target users. This means that there are no failures and that all functional and non-functional requirements are met. Software testing is a dynamic software quality management technique. In contrast to static software quality management techniques such as reviews, walkthroughs or inspections, dynamic techniques require the execution of the source code and the observation of its behaviour or outcome. One major advantage of such software tests is that, once implemented, their execution can easily be automated and repeated. This facilitates modifications of parts of the software such as extensions or re-implementations in later stages of the development, because repeatable software tests may be used to decide whether the software system still behaves as expected. The execution of software tests is an essential part of software development techniques such as Test-driven development.

Software tests can be categorized by two (orthogonal) properties: by their targets on the one hand and by their objectives on the other.

The objectives of software tests may aim to check one of various properties, which may either be functional specifications or non-functional properties. Some general examples of tests with differing objectives are: Acceptance tests, alpha and beta tests, Regression tests, Reliability tests, Performance tests, Security tests, Stress tests, Back-to-back tests, or Usability tests.

Regarding the targets there are three levels of abstraction: unit tests, integration tests and system tests. On the most detailed level unit tests focus on isolated modules or functions that may be tested on their own. They verify that the output of such a unit is correct given a selection of input values. Typically, but not necessarily, unit tests are conducted by the programmers who write the code. At the intermediate level, integration tests verify that interactions in groups of several modules work as intended and without errors. Depending on the system's architecture integration testing should be an incremental, carefully planned process, such that the involved software engineers can concentrate on one aspect at a time. At the most abstract level system tests are concerned with the entire software system. They are considered appropriate for assessing non-functional requirements, such as i.e. reliability, speed and security properties.

Tests at all three levels of abstraction are able to detect errors in the implementation. At unit test level and interaction test level, structural testing tends to be more suitable than functional testing, meaning that at these levels knowledge about the internal structure of the source code is used to derive test cases. However, at system level, functional testing methods tend to be more appropriate, which means that the system is treated like a black-box when test cases are derived. Therefore, system tests are typically better suited to validate that the software system as a whole meets the user's functional and non-functional requirements, whereas unit tests and integration tests are better suited to verify that the software behaves correctly and without errors.

Even in simple programs exhaustive testing is not always feasible because too many test cases are theoretically possible. Therefore, testing can only be conducted on a suitably selected subset of possible tests cases. There are many standardized test specification techniques that differ essentially in the way of how the set of test cases is selected, and software engineers must choose the most appropriate selection criterion based on the available knowledge about the system under test and the requirements and goals it should fulfil. In general, test specification techniques may be described as processes that follow a sequence of following high-level steps:

1. *Identification of test situations:* Each test specification technique aims at discovering certain types of errors. The first step in order to find these errors is always to identify situations in which they may occur. Therefore, depending on the nature of errors some source of information is required, which is referred to as the test basis. Structural tests for example require either the source code or a flowchart of the control flow as test basis and in this case all possible actions and decisions in the control flow are relevant test situations. In this step test specification techniques differ the most, because they have different requirements and different rules for the identification of relevant test situations.
2. *Construction of logical test cases:* Often a test case must be constructed as a sequence of test situations, because some test situations may only be reachable if other, preceding test situations had a certain outcome during the execution of a function from the beginning to the ending. Test cases for structural tests are for example created by combining all the previously identified actions and decisions to logical paths such that i.e. all branches of the control flow are covered and/or all outcomes of all decisions are produced, depending on the choices made by the software engineer regarding the intensity of the tests or in this case the code coverage.
3. *Concretion of test cases:* During the concretion of test cases their abstract representations must be mapped to concrete input values, steps to perform and expected results. In the example from above this means that a software engineer has to think of possible input combinations and possibly data to process, such that the control flow follows a specific path. The software engineer must also specify which results are expected. Depending on the complexity of the software system the choice of matching input values can be a tedious task.
4. *Preparation of initial data collections:* Often concrete test cases have preconditions or they

may depend on the system being in a certain state before they can even start. Sometimes for example a database must be reset to a certain state before the tests can be executed otherwise the outcome of the software might be unknown or less predictable. The required data is gathered in this step.

5. *Writing a detailed test script:* Finally, a detailed description of tasks in order to execute the test cases step-by-step is specified in the form of a test script. Beneath a sequence of actions to perform the test script also lists all preconditions and expected results.

The test script is the result of the test specification process. It is handed over to the software testers who then implement the actual software tests.

As mentioned earlier, there are different test specification techniques that can be utilized. They can further be categorized based on the way the test cases are generated: from the code structure, from the specifications, from real or theoretical faults to be discovered, from usage, from models or from the software engineer’s intuition and experience. The following list gives three examples of commonly used test specification techniques:

- *Code-based techniques:* These techniques use the source code as test basis. They may either focus on the control flow or on the data flow within the program under test. Control flow focused techniques aim to cover all statements of a program at least once and the overall code coverage is used to measure their intensity. Data flow focused techniques concentrate on the definition and use of variables in the program. The strongest criterion for such techniques is called *all-definition-use* and for each variable it requires the test cases to cover all control flow paths from the definition of that variable to the use of that variable. There are weaker forms of data-flow centred techniques that for example only cover uses or only definitions of the variables.
- *Input domain-based:* Input domain-based techniques focus on the generation of test cases that cover all variations of input values reasonably well. There may for example be a way to find equivalence classes among the inputs, meaning that one representative of each class may be chosen for a test case, because the behaviour or output of the program is the same for all elements of that class. There are also techniques that systematically generate combinations of interesting input values. Other techniques choose input values near the boundaries of the input domain and others just generate random test cases.
- *Model-based testing techniques:* These techniques rely on an abstract or formal model of the software under test in order to choose test cases with a focus on the behaviour of the software. Decision tables for examples are a technique that may be used in order to consider all possible combinations of conditions and their corresponding resulting outputs. Finite–state machines may be used as a model of software system in order to cover all states and state transitions with test cases.

5.1.4.2 Test-Driven Development

Test-driven development (TDD) is a software development process that relies on the repetition of a very short development cycle: requirements are turned into very specific test cases, and then the software is improved to pass the new tests, only. This is opposed to software development that allows software to be added that isn't proven to meet requirements.
The method can be summarized in several steps [19]:

1. Add a test: A test should define a function or improvements of a function, which should be very succinct.
2. Run all tests and see if the new test fails: This step verifies the validity and necessity of the new test. The new test should fail for expected reason.
3. Write the code: The added code (may not be perfect) should cause the test to pass.
4. Run tests to see if all the test cases now pass. This step wants to verify that the added code does not break or degrade any existing features.

5. Refactor code and remove the possible duplication.
6. Repeat the cycle to push forward the functionality.

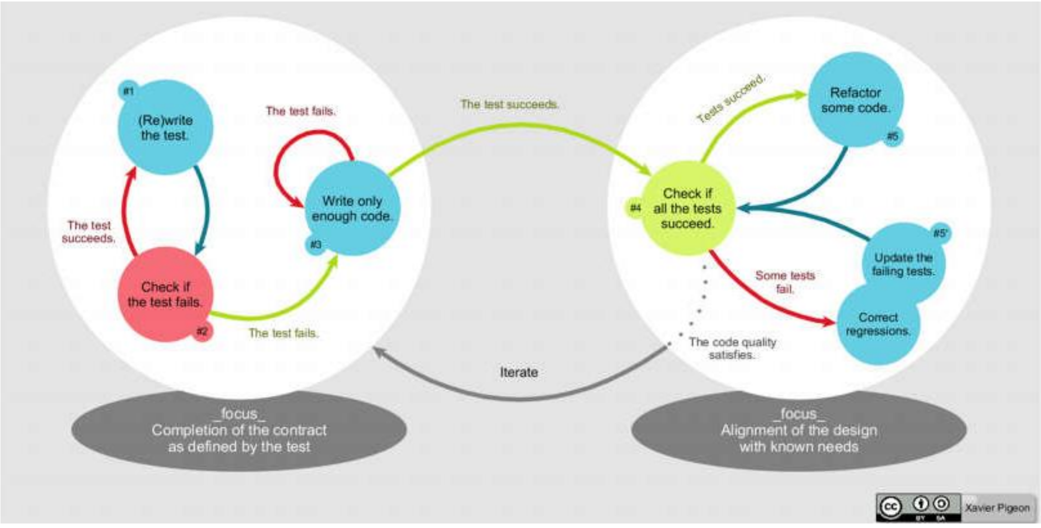


Figure 5.8: Test-driven software development cycle

5.1.4.3 Domain Independent Test Specification with TTCN-3

While there are various testing procedures in the literature, for physical systems as well as software implementation, they are mainly domain specific approaches and do not taking into account interaction with other domains (e.g software vs. physical). Besides, even though the existing testing procedures are quite classical, the actual deployments differ and are the choice of vendors. These two reasons lead to:

- Costs on test training and maintenance.
- Difficulty in comparison of tests carried out by different vendors.
- Necessity of conducting an integration test after domain-specific tests.
- Difficulty in conducting large and complex multi-domain tests.

Alongside with the integration of IT developments, modern engineering systems (e.g electrical grid, mechatronic system or SCADA) have become complex juxtapositions of technological domains, it is necessary to standardize a testing technology that is independent from domains of application. Moreover, the standard should not be neither tied to a particular application or its interfaces, nor tied to any specific test execution environment, compiler or operating system. In this context, the European Telecommunication Standards Institute (ETSI) has developed and has been maintaining The Testing and Test Control Notation Version 3 (TTCN-3), specifically for testing and certification. It has been also considered in the mandate M/490 [11] in the context of smart grid system interoperability.

The ETSI TTCN-3 standards have also been adopted by the International Telecommunication Union (ITU-T) in the Z.160 series. The series consists of several standards in the package ES 201 873. TTCN-3 is:

- A programming language specifically designed for black-box testing and certification.
- Applicable to a variety of domains and types of testing.
- Adapted to very large and complex industrial tests.
- Is not executable and requires a compiler/interpreter, adapter as well as codec implementations.

In general, TTCN-3 describes at an abstract level (similar to CIM for Transmission/Distribution grids). The standard specifies tests, but a corresponding test system is needed for test execution. That structure can be represented as in Figure 5.9 [67].

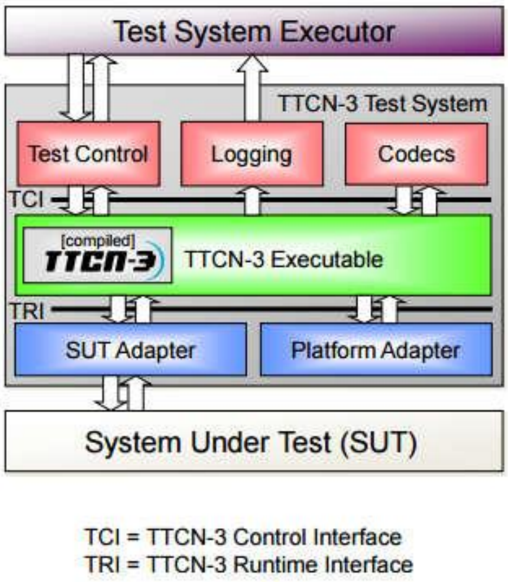


Figure 5.9: TTCN-3 Structure

TTCN-3 is used mainly to automate conformance and Interoperability testing. As a language at abstract level, TTCN-3 can be integrated with systems in other languages (ASN.1, XML, C/C++). Depending on the situation and necessity, TTCN-3 can involve minimal or concurrent tests.

The TTCN-3 standard provides templates, syntax and vocabularies to define a conformance and interoperability test. The applications are extensible to unifying testing technology in engineering. However, as a high-level platform independent language, no clear advantage of TTCN-3 over other meta-languages is recognized.

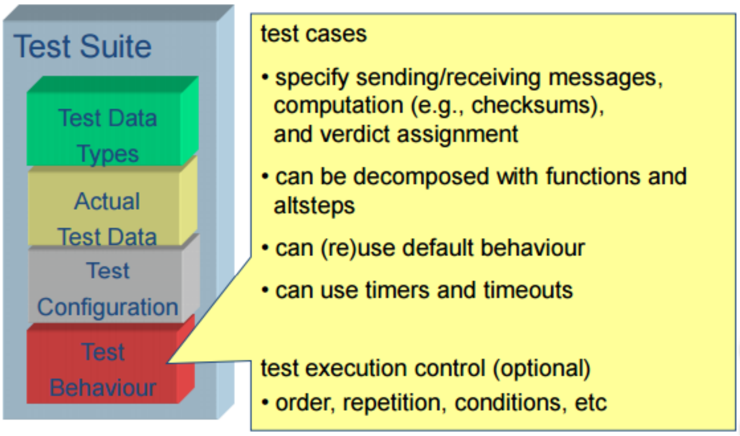


Figure 5.10: A test suite in TTCN-3

TTCN-3 has been widely used in testing telecommunication equipment and software since 2000. The electrical community has recently begun to adopt TTCN-3 to automate interoperability tests [68]. While the applications of TTCN-3 in the electrical domain testing is still limited, its concept of abstract testing layer is promising in the development of holistic testing procedure.

5.1.5 Relevance and Conclusion

The current practices at the RI involved in ERIGrid give information about the different tests conducted, technologies used and degree of automation. In perspective of a holistic test among different RI that are not physically coupled this gives first indications of information needed to decide what test capabilities are compatible. These findings should be included in a test case description method. Some of the information in the questionnaires has been ambiguous and thus an unambiguous classification of test cases to clusters has not always been possible. For this reason, clear definitions are necessary. Furthermore, different levels of detail for the descriptions are needed in order to have general information about test cases that then can be inferred down to information used at specific RIs. In this section, first the results of the current practices are discussed followed by a discussion of the state of the art with regards to their relevance for the ERIGrid purpose.

One result from the current practices questionnaires was that a clear separation of the object under investigation and the system is necessary in order to describe a test and to be able to make dependencies explicit. It becomes clear that different domains are involved in the tests. The investigations regarding the purpose of investigation made clear that it is difficult to separate the purpose from the objects. However, a clear description might help to identify dependencies and interrelation between these. Test setups identified were hardware tests, real-scale components but also grid or component simulators or emulators. Furthermore, procedures adopting Real Time and Hardware-in-the-Loop methodologies were identified, in total giving a wide range of technologies involved and the possibility for realising tests of a similar purpose in different ways.

One addition to the investigations could be to include synergies of energy networks. That would depend on various factors and especially the overall project objectives, which may focus on more than one energy domain. In principle, this is particularly important when storage technologies are considered, since several of the testing procedures in the consortium involve storage applications. On the other hand, the involvement of these technologies in the specific testing procedures is indirect with emphasis on the impact that their implementation has on the electric domain rather than the evaluation of the storage characteristics itself. The idea of incorporating a domain-classification has also been introduced by the test criteria working group. A relation to target parameters or metrics will have to be added and specified.

In the test design working group a general test flow and classes of different test flows were identified. This is valuable information regarding the input-output relations of one test or experiment but also the number of runs necessary to obtain a certain quality.

Investigations about possible gaps regarding interoperability and protocols in use showed that the exact domains' boundaries will have to be specified in more detail, because this is how inter-domain connections can be accurately defined. The focus in the partners' RIs lies on connections between the ICT and electrical domain. However, there should be other inter-domain connections, such as physical means or processes like conversion of electrical to thermal energy etc. Apparently, it is much more difficult to specify other than ICT interconnections because they may be complex systems, abstract processes etc. Also, the ICT interconnections are of higher importance because they can allow for the virtual connections. Hence, the questions arise if and in which degree of detail boundaries of domains must be described in order to identify the inter-domain connections and if other inter-domain connections that regard other than the ICT domains are necessary.

Different protocols are widely used at the RIs. However, the underlying data models might not be compatible if a specific data structure for storing results is used at all. A physical connection is probably not feasible, however a "virtual" or "offline" connection might be possible.

A possible framework for exchange of experiments results might be obtained by the metamodelling methodology. The literature study has shown that meta-models are being used in different domains and different problem sizes. Thus, using the design of experiments approach to generate meta-models can be considered a potential methodology for representing tests at different RI. Metamodels could be passed from one RI to another, such that the execution of combined tests becomes possible without physically coupling the RIs. However, the feedback from the partners have shown that in order to use metamodels in practice the process has to be defined in more detail covering at least the preparation and exchange steps. From the use case and function under investigation the requirements of metamodels regarding input and output parameters as well as the model accuracy or quality must be derived. Furthermore, interfaces for model exchange must be specified. As indicated above, components from the electrical and ICT domain form the majority considered in the partners' RIs. Thus, those have been investigated in more detail.

The complexity of a test depends on the purpose and object or system under investigation. Whereas for component testing open-loop testing suffices, this is not the case in power system testing since it is system level testing method applied on a single domain. The connection to other (physical) domains is the next step. To integrate different RI for tests, the different levels or hierarchy of tests, their properties as well as interconnections must be taken into account.

In the area of software testing different levels or hierarchy of testing occurs as well: unit tests, integration tests and system tests. A mapping between these levels of abstractions in different domains could be especially relevant for inter-domain testing. Especially methods for testing in the context of hardware-software interfaces such as for embedded systems may help in the mapping process.

The *Holistic* meaning of test-driven development is the validity of all the tests, or in another way, all the features are re-considered once a new feature is added. The ERIGrid *Holistic testing* may correspond to the *Test suite* concept in TTCN-3. The lack of domain specific vocabularies may be overcome with a combined usage of TTCN-3 and CIM. ERIGrid test cases may require a much more complex taxonomy. However, the idea of constructing an abstract layer for test case description, with its proper language is relevant to the project.

5.2 Holistic Test Case Description

In this section, we describe in detail the holistic test case formulation.

5.2.1 Motivation for a Holistic Test Case Description

A test description aims to clarify the object under investigation, test objective, and by what means a test is to be carried out (i.e., test system, setup and test design): *1. what* needs to be tested, *2. why, and 3. how.*

As outlined above, the holistic testing procedure (Section 2.3.2) envisions a separation of the first two pillars of a test specification (i.e., test object and test objective) from the third (the means of testing). We refer to a holistic test case as specification of the *what* and *why* of a test, without including specific limitations on test setup and test design. In contrast to conventional power systems testing, this requires a more formal approach, as the intention of a test case must be unambiguously identifiable, enabling specification of a test design, and test/experiment setup in a separate step. Another aspect of the holistic testing approach is the merging of different cultures of testing, which can be portrayed as a device-oriented culture of physical testing and a culture of testing ICT objects such as implementations of protocols and algorithms. Rigorous formal specification of test cases as well as automated execution of tests are common in the ICT domain (as explained in Section 5.1.4). In the testing of physical components, the test object is delimited by its physical boundaries, requiring little further formalization of the test object. However, a good test specification requires insight on physical and engineering principles. Test specifications therefore tend to be domain specific and less formal. Further, much of the test design is decided by the available test setup. A challenge is therefore to formalize the complete cyber-physical system context and test criteria, to formulate a test case combining several ICT and physical components and sub-systems as well as test criteria spanning different domains. The holistic test case identifies and formulates this required formal context of a test specification by assembling the domains and component types, breaking and structuring test objectives so that clear test criteria are formulated.

5.2.2 Test Case Methodology & Definitions

As illustrated in Figure 2.1(a refinement of Step (1) in Figure 2.4, page 28) we envision the specification of a holistic test case as composed on the basis of the following description items: a smart grid scenario composed on the basis of a generic system configuration (SC) and related use cases, as well as the intention of a test objective summarized in a *narrative*.

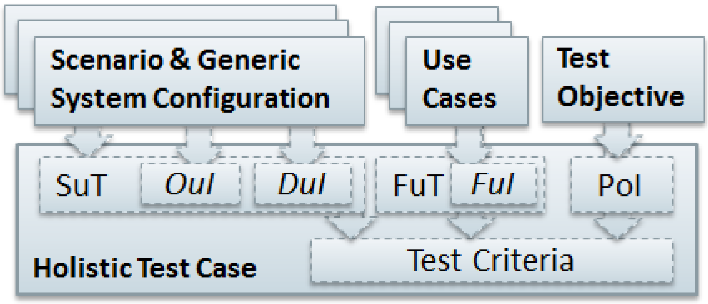


Figure 5.11: Elements of a holistic test case

Here, the System under Test (SuT) identifies the abstract, categorical, system boundaries of an abstract test system encompassing all relevant sub-systems and interactions (domains) required for the investigation. The Object(s) under Investigation (OuI) identify the subsystem(s) or component(s) in scope of the test objective, and with respect to which the test criteria need to be formalized. The Domain(s) under Investigation (DuI) identify the relevant physical or cyber-domains of test parameters and connectivity, which can be listed directly or formulated in form of a hierarchy (cf. Section 3.2.3.2, page 41). With reference to use cases, the full set of Function(s) under Test (FuT), and the specific Function under Investigation (FuI) are identified. The Purpose of Investigation (PoI) formulates the test objective, also stating whether it relates to characterization, validation or verification objectives. Together the above items inform the Test Criteria, which formalize the test metrics into: target criteria, variability attributes, and quality attributes (which are typically thresholds for acceptable results).

When analysing the test objectives to formulate the PoI, note three qualitatively different types objectives of a test:

Test objectives: The purpose for carrying out the test. These can be divided into three categories:

- *Characterization*: a measure is given without specific requirements for passing the test. Examples: understanding the behaviour of a system, developing a mathematical model of a component.
- *Validation*: a requirement and abstract measure is provided, but the results are subject to interpretation, i.e. passing a test depends on a qualitative evaluation by an expert or user of the system. These tests seek to answer the question *are we building the right system?* Example: Is the mathematical model good enough?
- *Verification*: acceptance of a test result depends on the direct evaluation against fixed and formalized assessment criteria. These criteria can be formulated as quantitative measures with a set/range of acceptable values of these measures, i.e. quantitative tests. These tests seek to answer the question *are we building the system right?* Example: Testing whether a component conforms to a standard.

These concepts have been derived from the questionnaires mentioned in Section 5.1.1. Naturally, “characterization” tests have a more important role in earlier stages of systems design, for example associated with prototype or concept testing. Validation tests are typical tests in the higher stages of development, typically toward the end of a V-development process, where the result of more complete test settings have to be interpreted to assess, for example, whether customer requirements have been satisfied. The *verification* type of test objectives corresponds to tests that are relevant to standardization, as the earlier discussed *compliance* and *conformance* tests. Interoperability testing following the definitions provided in [11] is a *verification* assessing the conformance of a device with the required interoperability profiles.

Definitions of Test Case elements

Although the following terms have been used throughout the some of the previous sections they are repeated here in order to provide their definition in this common context them so that the section can be used for future reference.

Test Case: A test case provides a set of conditions under which a test can determine whether or how well a system, component or one of its aspects is working given its expected function.

System under Test (SuT): is a (specific) system configuration that includes all relevant properties, interactions and behaviours (closed loop I/O and electrical coupling), that are required for evaluating an OuI as specified by the test criteria.

Object under Investigation (OuI): the component(s) (1..n) that are to be characterized or validated.
Remark: OuI is a subset of the SuT.

Domain under Investigation (DuI): Identifies the relevant domains or sub-domains of test parameters and connectivity."

Functions under Test (FuT): the functions relevant to the operation of the system under test, as referenced by use cases.

Function(s) under Investigation (FuI): the referenced specification of a function realized (operationalized) by the object under investigation.
Remark: the FuI is a subset of the FuT.

Purpose of Investigation (PoI): a formulation of the relevant interpretations of the test purpose (e.g. in terms of Characterization, Verification, or Validation)

Test criteria: the measures of satisfaction that need to be evaluated for a given test to be considered successful:

- *Target metrics* (criteria): A numbered list of measures to qualify (quantify) each identified Purpose of Investigation.
- *Variability attributes* (test factors): identification of the sets of attributes (controllable or uncontrollable parameters) and qualification of the required variability; includes reference to purpose of investigation.
- *Quality attributes* (thresholds): with reference to purpose of investigation and/or target metrics, the threshold level required to pass a test or the certainty/precision level (e.g. probabilistic measure) required for the quality of a characterization.

As illustrated in Section 3 and below, the SuT, DuI and OuI identification can be directly associated with a generic system configuration, the Test Case Context model or (TC-GSC), cf. Table 3.2: Classification of System Configuration Types (SCTypes). This generic system configuration is the background for defining any of the required *test systems (TS-SC)* for the sub-test specifications derived from this test case.

5.2.3 Step-by-Step Guideline for Holistic Test Case Template

The definition of a holistic test case entails the following steps:

1. Motivation and context of Test case: Set scope and goal:
    a. Formulate the narrative in one sentence or paragraph:
        i. Test case or test objective?
        ii. To what use case does it apply? in context of what system configuration?
        iii. Define a unique test case identifier (if relevant)
    b. Identify related Generic System Configuration (GSC) and Use Cases (UC).
    c. Revisit the test objective to ensure it is stated in relation to the GSC and UC elements.

2. Identify Holistic test components:
    a. Identify the System under Test (SuT) within the Generic System configuration
        i. If not explicitly identified here, any component of the SuT may become OuI in the following specification steps
        ii. The domains identified in the SuT are all possible Domains under Investigation, unless the DuIs are identified further here.
    b. List the functions:
        i. FuT: functions required to be operational in the SuT
        ii. FuI: functions for which test criteria have to be defined.
    c. Purpose of Investigation (PoI):
        i. Reformulate test objective into a numbered list (PoIs) so that at least one objective is specified per expected test.
        ii. Ensure that each PoI is formulated wrt. A specific OuI and/or FuI
        iii. Ensure that each PoI is qualified as either characterization, validation or verification.

3. Specify *Test criteria* for each PoI (reference PoI list items)
    a. Formulate the *target metric* as a quantity to be derived from SuT and DuI related variable types.
    b. Identify *variability attributes* qualitatively as ranges of relevant test parameters in terms of acceptable uncertainty and required variability (also) for non-OuI components of the SuT.
    c. Define the quality attributes, for assessing an acceptable test result. In case of a *characterization* PoI, here the remaining model uncertainty is stated; for *verification* PoI, the acceptance threshold (worst case for passing the test) is stated; for *validation* PoI another criterion for ending the test execution can be chosen.

These steps are clarified through the three examples in the two following sub-sections. If distinctions between test case and use case or system configurations are unclear, please refer to the mini-tutorial provided in Annex 9.7.

5.2.3.1 Example 1: Inverter Loss of Mains

This section provides a simple example for developing a component test case. This is based on actual testing defined and performed on low voltage PV inverters. This will be used as stepping stone for developing more elaborate holistic test cases. This should be considered in conjunction with the test case template provided in the report appendix.

Motivation of test case:

Prior to identifying a use case and related system configuration necessary for developing a test case, the objectives of performing the test should be defined. These objectives formulate the purpose of investigation (PoI).

In this particular example, it is necessary to verify whether PV inverters supplied by different vendors are able to detect true islanding (loss of mains, LoM) conditions and as such trip the inverters accordingly. Furthermore, to verify if the inverters are capable of riding through large scale grid disturbances manifested as grid frequency excursions with relatively high rate of change of frequency (RoCoF) values. This investigation focuses on the direct influence of inverter built-in LoM protection functions on the two aforementioned grid operating conditions. It is required to conduct the tests at this stage using physical inverters because models of inverter controls do not necessarily reflect the real behaviour during these conditions. This requirement will have a bearing on the experiment specification.

In order to frame this motivation in a test case, it can be said that the PoI is to experimentally *verify* the sensitivity and stability of *PV inverter* loss of mains (LoM) *protection functions* under true islanding and frequency stability conditions respectively.

In this particular example, the main objective of testing is that of verification with respect to the performance of a function – that is inverter built-in LoM protection functions. The impact of the performance of these functions of interest is considered at two different levels, which has a bearing on the applicable system configuration and use cases. On one level, the sensitivity of the inverters to islanding conditions and the requirement for them to trip during this condition is of relevance to the local distribution network where the inverter is connected. On the second level, the requirement for the inverters to remain stable during grid frequency excursions has an impact on the grid stability especially if a large number of inverters are incapable of riding-through said disturbance.

Identify System Configuration and Use Cases:

Generic system configuration

As discussed in the motivation section, there are two levels at which the impact of the inverter built-in protection functions performance should be considered. At first glance this may indicate the requirement for two separate system configurations (i.e. ‘distribution grid’ when considering the impact on the local grid and ‘vertical integration’ when considering the impact on the overall grid stability). This is not the case, however, when considering the interactions of interest while conducting the test. In this example the only test outcome of interest will be whether the inverters trip or ride-through in each grid operating condition. The interactions of interest will be clear when the system under test (SuT) is defined with its constituents, while the test outcomes will be formulated as test criteria.

The next step following from the system configuration identification is to perform a mapping into a system under test (SuT).

Use case description

Single and three phase PV inverters are connected to a low voltage (LV) distribution network. These inverters are monitoring their terminal voltage to perform LoM protection functions which trips the inverter if a distribution island is created. The distribution grid voltage and frequency is affected by local and remote faults and grid instabilities.

The next step following from the identification of the use case is to define the FuT.

Formulate test objective:

As described earlier, the PoI is to verify the performance of inverter built-in LoM protection functions under different grid operating conditions. This can be formulated into two test objectives:

Test objective I

Verify that the inverter LoM protection is sensitive to islanding conditions with a local load-generation (generation provided by the inverter under test) mismatch of up to 1% of the inverter rating.

<u>Test objective II</u>

Verify that the inverter LoM protection is stable to grid frequency disturbances of up to 1Hz/s.

Define Holistic test components:

In order to define a SuT, a mapping from the generic system configuration to specific components is required. The figure below shows the SuT for this particular test case, which is based on a distribution grid system configuration. This SuT encompasses a low voltage distribution grid, points of common coupling (PCC) used to define the boundary of the island, loads and PV inverters, which embody the OuI. The only domain of interest in this SuT is the physical-electrical domain, which is the DuI.

As described earlier, the verification test is performed on the inverters' built-in LoM protection function and these represent the FuI. In order to realise the use case as described earlier, a number of functions need to be represented in the tests (i.e. FuT), these are:

- Grid frequency control functions (these have a bearing on the grid frequency and RoCoF during a disturbance).
- Network protection functions (this will influence the change of state of the PCC).
- Demand control functions (this influences the power flows in the LV distribution network).
- Network voltage control (this influences the voltage levels at different LV busses as well as the network power flows).
- Inverter protection functions (these enable connection of the inverters to the grid as required by the relevant grid codes).

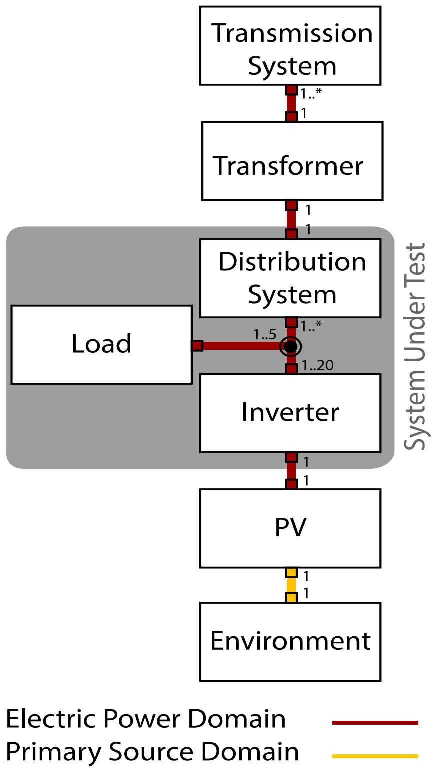


Figure 5.12: Illustration of the SuT with reference to a distribution grid system configuration

It is important to note that FuI are not necessarily realised in detail when performing the actual experiment. However, the domain attributes such as pre-event power flows and voltage levels should be defined in the test specification and realised in such a way that satisfies the PoI.

Refine specific Test criteria for each PoI:

In order to perform the verification testing, measures of satisfaction for each PoI is required. Note that the test criteria are not restricted to the FuI and OuI, but it should reflect the SuT and FuT for each defined PoI. These measures of satisfaction are quantified/qualified as follows with respect to this example:

Target metrics

- To verify inverter sensitivity to LoM events, the target metric of interest is the connection status the OuI. In order to satisfy the PoI, the 'connected status' of the OuI should be 'false' (i.e. the inverter is tripped).
- To verify inverter stability to grid frequency disturbances, the target metric of interest is the also the connection status of the OuI. However, in order to satisfy the PoI, the 'connected status' of the OuI should be 'true' (i.e. the inverter does not trip).

Variability attributes

A set of controllable and uncontrollable parameters are involved in the test. Those that are directly controlled with respect to the OuI are:

- Grid rate of change of frequency for verifying the FuI stability, while remaining within the FuI under and over frequency protection functions trip thresholds.
- Pre-islanding power flow across the PCC for verifying the FuI sensitivity.

Quality attributes

The test metrics must qualified/quantified in order to assess whether the test criteria are satisfied. This qualification/quantification encompasses both the thresholds that the metrics should be bounded by (which represents a test outcomes) as well as the quality of observed metrics (which represents how well the test outcomes could be used for this assessment). In this case:

- The FuI issuing a binary trip command to disconnect the inverter (OuI).
- Stability of the FuI for a grid frequency disturbance of up to 1Hz/s.
- Sensitivity of the FuI for a pre-islanding power flow across the PCC of up to 1% of the OuI power rating.

5.2.3.2 Example 2: Voltage Control

This example illustrates the difference between a classical component test and a system test by describing each of them in the terminology of the holistic test approach.

Component test example

An example case for the component test introduced in Section 5.1.2 is explained here. The SuT is the PV inverter, which coincides with the OuI. The DuI is electric power. The FuT coincide with the FuI and can be the MPPT operation, the FRT capability, the Q(U) controller etc. The test objective, or PoI, is the verification (e.g. according to standards) of the aforementioned inverter's functions according to certain testing criteria, such as the efficiency of the MPPT, maintaining the operation at specific voltage dips (for FRT) and the reactive power provision (for Q(U) control) respectively.

An example of a FRT test is shown in Figure 5.13.

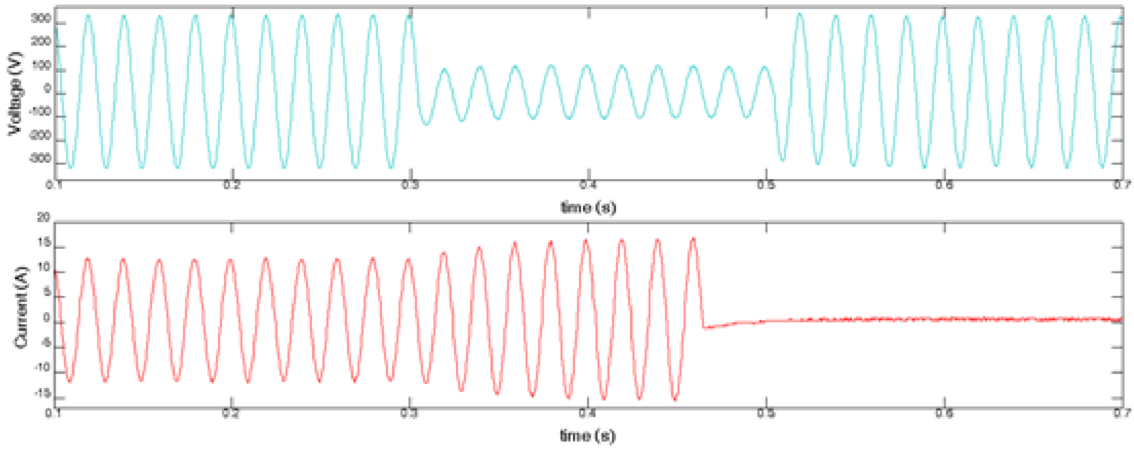


Figure 5.13: Voltage dips performed on a commercial PV inverter.
A predefined voltage profile is applied (i.e. open-loop test)

System test example

An example case for the power system test introduced in Section 5.1.2 is presented here. The SuT comprises the whole system i.e. PV inverter, OLTC, transformer, distribution line and upstream network impedance. The DuI is electric power. The OuI are both the OLTC and PV inverter, as interactions between these two components are examined. The FuT and FuI are the Q(U) droop controller of the PV inverter and the OLTC controller which regulates the voltage of its secondary winding. Both FuI are implemented as local controllers. The test objectives-PoI are the characterization and validation of the SuT. The test criteria [P1] are the OLTC reaction to reactive power levels and inverter's reaction to tap changes. Interactions between the two controllers within the power system are shown in Figure 5:14 [55].

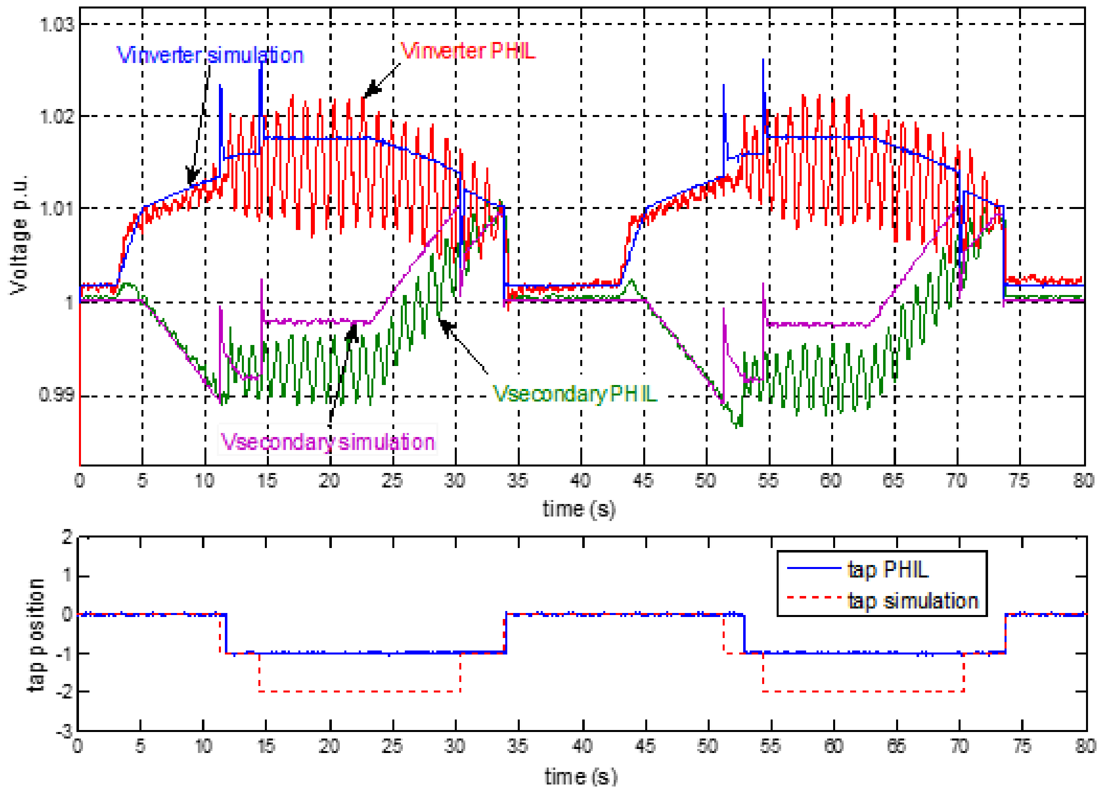


Figure 5:14: Power system testing: Interaction of OLTC and PV inverter
(PHIL testing and pure simulation results)

In order to examine the overall system behaviour in complex system configurations, holistic tests should be performed which include multiple domains, e.g. electric power and ICT.

One example is the characterization and validation of a Distribution Management System (DMS) controller (i.e. central voltage controller) which monitors and controls several devices of an active distribution network as shown in Figure 5.15, where a generic system configuration is presented.

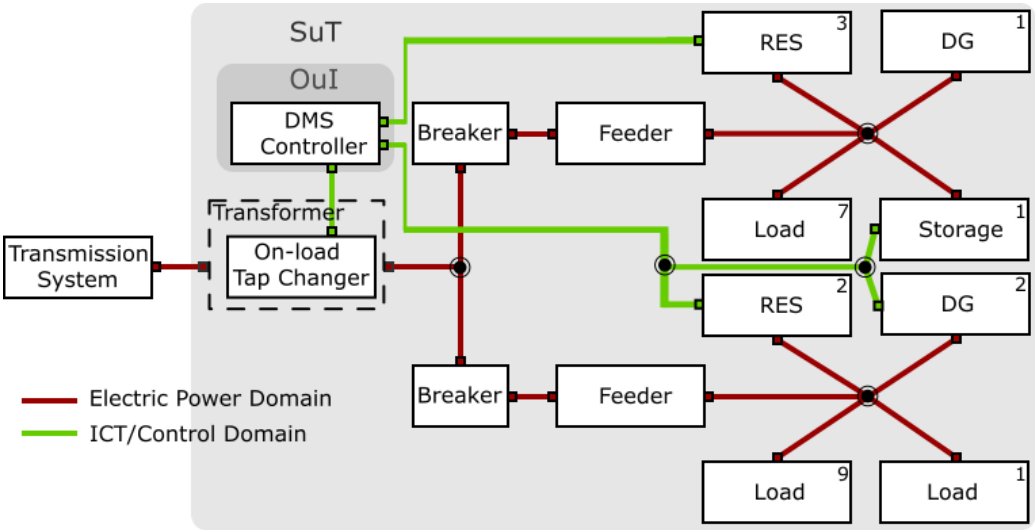


Figure 5.15: Holistic testing of DMS controller and active distribution network

In this example the SuT includes the DMS controller, DER devices, OLTC controller, transformer, distribution lines (from the electric power domain) and telecommunication network (from the ICT domain). The OuI is the DMS controller, which is the main point of interest and the DuI are both the electric power and ICT domains. The FuT are the state estimation and optimization of the DMS controller, the DER P-Q control, OLTC tap control, measurements, and communication via ICT. The FuI are the state estimation and optimization executed in the DMS controller as they are functions of the OuI. The test objectives-PoI are the DMS controller characterization and validation in terms of:

1. Convergence of the optimization (validation)
2. Performance of the optimization under realistic conditions (characterization)
3. Accuracy of the state estimation (characterization)

The test criteria are divided in target criteria, variability attributes, and quality attributes and are described below:

- *Target Criteria* (according to the aforementioned test objectives)
  1. When and how often the optimization converges. How fast and what is the solutions quality (suboptimal, etc.)
  2. Voltage deviation of all the nodes from the nominal value, number of tap changes, network active power losses
  3. Estimation errors of voltage, active and reactive power
- *Variability attributes*:
  1. Load and RES Patterns (realistic, daily, annual variation), communication attributes (packet loss, delays)
- *Quality attributes* (thresholds):
  1. Convergence within 2 sec (validation)
  2. All voltages measurements have precision of ±2% of the nominal value (characterization)
  3. Estimation quality characterized with a confidence of 95% (characterization)

A comparison with two more conventional test cases, as referred in the state of the art section, is provided in above. The table illustrates that the proposed methodology can well be applied to conventional tests cases, but also clearly marks the difference between the three referred test types.

Table 5.2: Component, Power System and Holistic Testing

Element	Component Testing	Power System Testing	Holistic Testing
SuT	PV inverter	PV, OLTC, transformer, distribution line, upstream network impedance	DMS, DER, OLTC transformer, distribution lines, telecomm network
DuI	Electric Power	Electric Power	Electric Power, ICT
OuI	PV inverter	OLTC, PV inverter	DMS controller
FuT	MPPT, FRT, Q(U) control	- PV inverter's Q(U) droop control -OLTC controller regulating the voltage of secondary winding	State estimation and optimization in the DMS controller, DER P-Q control, measurements, OLTC tap control, communication via ICT
FuI	MPPT, FRT, Q(U) control	- PV inverter's Q(U) droop control -OLTC controller regulating the voltage of secondary winding	State estimation and optimization in the DMS controller
Test objectives	Verification of MPPT, FRT capability, Q(U) control	Characterization and validation of the SuT	Characterization and validation of the DMS controller

5.2.3.3 Example 3: Aggregator Validation

An aggregator, controlling DERs in 500 households, wishes to participate in the ancillary service markets by providing secondary frequency control to the transmission system operator (TSO). This holistic test case is a part of pre-qualification tests an aggregator must pass [21] in order to participate in aforementioned markets. Figure 5.16 presents the general system configuration. In this test case it was analysed how the aggregator control system tracks the Automatic Generation Control (AGC) signal supplied by the TSO when subjected to disturbances in its ICT infrastructure. Metering issues and impact on the distribution grid are out of scope of this specific test. The holistic test case is described by:

- PoI: to characterize the sensitivity towards ICT disturbances of the ancillary service quality of an aggregator.
- SuT: the system under test is composed of the aggregator infrastructure and 500 households. The input to the system is the AGC signal sent by the TSO and the output is the power consumption/production of the households.
  - OuI: aggregator control system (part of the aggregator infrastructure).
  - DuI: electric power and ICT infrastructure.
- FuT: Aggregator central control, local DER control, communication functionality between aggregator and home energy management system (HEMS).
  - FuI: Aggregator central control.
- Test Criteria:
  - Target criteria: Service quality, measured as the difference between the reference signal from the TSO and the aggregated power consumption/production of the DERs as measured by the individual DER measurement systems.
  - Variability Attributes/Test Factors: The ICT connection between aggregator and HEMS.

- Quality Attributes/thresholds: The ICT parameters are to be varied until the aggregator is unable to track the AGC according to the contract.

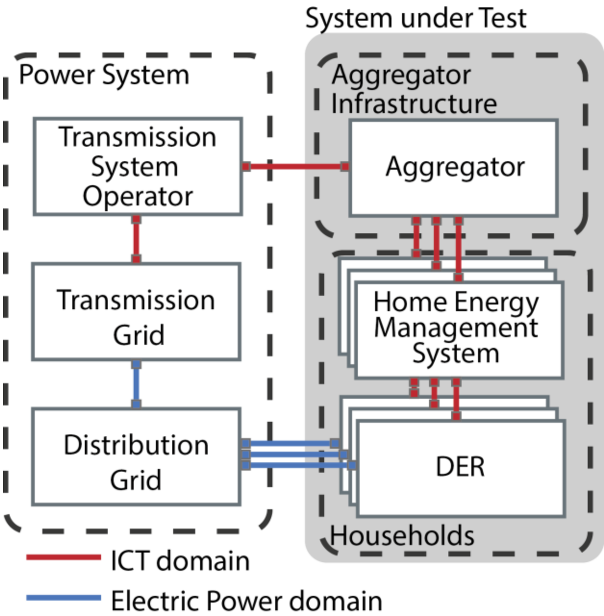


Figure 5.16: The test setup described by the component centric approach. Note that the aggregator infrastructure and households form the SuT

5.3 Test Specification

Whereas the test case defines the context and objectives of a holistic test, the test specification defines the actual test system. The test specification results from interpreting a single purpose of investigation of a test case and mapping the implied test requirements to specific test system. If a holistic test case entails several sub-tests, one test specification is written for each sub-test.

5.3.1 Motivation

A test specification aims to clarify the object under investigation, test objective, and by what means a test is to be carried out (i.e. test setup and test design): what is to be tested, why, and how. This specification is derived from the holistic test case and may comprise tests covering components or a sub-system of the (holistic) system under investigation as a result of the mapping step (as outlined in Section 2.3). Thus, based on the holistic test description, the test must be mapped to suitable RI and there possibly separated into sub-tests covering sub-systems or components of the holistic system. Hence, the test specification should reflect information to be realised at a (not specific) RI. During the discussions and developing the description template relevant information from the partner's current practices have been considered and iteratively extended.

5.3.2 Methodology & Definitions

In the test specification template (see Appendix Section 9.6.2) information is gathered that enables the assignment of a test to a specific infrastructure but also specifies requirements for a test to be conducted. The term system under test may refer to a sub-system or component specified in the holistic test. Accordingly, the object under investigation may refer to one or more of the OuIs in the holistic test specification. The information is given below as well as corresponding definitions and explanations.

Test System: defines how the object under investigation is to be embedded in a *specific system under test*, which parameters of the system will be varied and observed for the evaluation of the test objective. Objects and functions that are not part of the system under test are simplified to the minimum function, functional equivalent or boundary parameters necessary to execute the test.

Input and output parameters: Here, a list of inputs for the system under test relevant to the object under investigation, inputs relevant to the object under investigation itself are given. The inputs are divided into the categories of controllable and uncontrollable parameters. Furthermore, a list of outputs respectively measured parameters is given.

Test Design: Based on the test criteria defined in the test case, the test design specifies how the required targets metrics are derived or generated from the input parameters and measured parameters. This may be specified as a sequence of steps, variation of (controllable) input parameters or decision criteria. This can be given as textual or graphical description of the sequence of steps carried out during the test including parameter ranges and variation or choice of the input parameter.

Initial system state: The initial system state is described as conditions that are prerequisites to actually run the test and initial choices of parameters.

Evolution of system state and test signals: The temporal evolution of test events and evolution of the relevant test parameters are quantitatively characterised as they may be adjustable by the input parameters (e.g. opening breakers after a certain amount of seconds). Furthermore, the evolution of the variability attributes over time is given.

Other parameters: There may be information of data that should be tracked apart from the input and output parameters and system state, test signals. These parameters are listed here.

Storage of data: The format in which the parameters are stored is given as well as the technology that is used.

Temporal resolution: The temporal resolution is given as discrete or continuous and (if applicable) the resolution of the discrete time steps is specified. This is especially interesting for simulations.

Source of uncertainty: In order to evaluate the quality of the test, the possible sources of uncertainties are given and how they can be quantified.

Suspension criteria / Stopping criteria: A list of conditions should be given under which the test results are decided to be either valid or not valid, or the test is interrupted.

5.3.3 Example (Step-by-step Guideline)

The following example is a continuation of the example presented in Section 5.2.3.1.It exemplary shows the usage of the information introduced before and how it is used to develop a test specification.

Test system:

The diagram in Figure 5.17 illustrates the example test system. It belongs to a distribution grid system configuration where PV inverters are connected to a low voltage AC distribution network. The OuI's are depicted within the dotted green line. At this stage, aspects of the SuT are described more concretely such as the voltage levels the scope of the distribution grid and the general topology. While Figure 5.17 shows a complete grid topology using the notation presented in Section 3, it can conveniently be simplified to the diagram presented in Figure 5.18. Note that the terminal names are the same, and the system topology could be represented by a domain specific diagram, e.g. a line diagram, with correct labelled components.

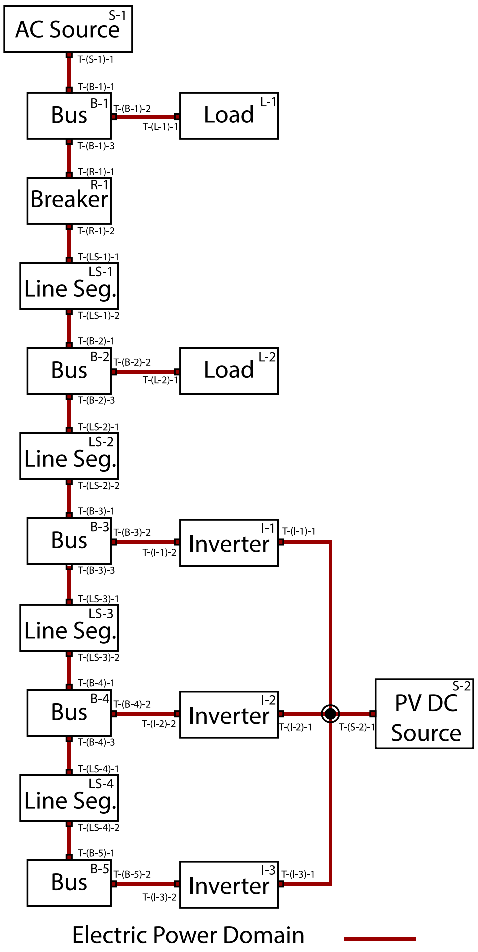


Figure 5.17: Example test setup for inverter testing

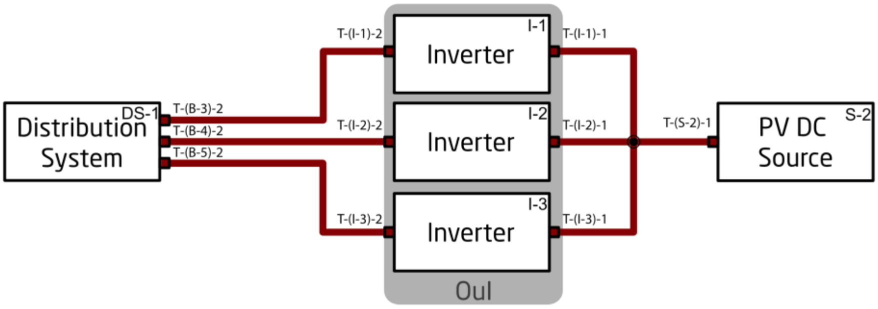


Figure 5.18: Simplified Test System Configuration for the inverter-testing example

Input and output parameters:

The following inputs are relevant to the OuI or the SuT:

<u>Controllable input parameters</u>

- Inverter DC power input.
- Grid frequency.
- Grid voltage.
- Grid loading.
- Inverter regional settings (e.g. G59).

Uncontrollable input parameters

- Inverter reactive power output (not controllable directly).
- Inverter controller settings (depends on the inverter).

The following outputs are relevant for the test:

Measured parameter

- Inverter AC terminal current and voltage (derive frequency, RoCoF† and power).
- PCC voltage, current and breaker position (derive power†).
- Inverter connection status (trip/no trip)*.

Target metric (i.e. inverter connection status given a RoCoF value for stability verification tests and inverter connection status given PCC power for sensitivity verification tests) as described in the test case

Variability attributes as described in the test case.

Test Design

The following is considered when designing the test:

- Inputs to stimulate the FuI (maximum grid RoCoF of 1Hz/s).
- Means of varying power flow through the PCC to create a true loss of mains condition (between 0-1 % of inverter kVA rating).
- Network topology and hence reactance to minimise reactive power flow (e.g. shortest cable, overhead line lengths).

The test design is motivated by the inputs to the FuI. In this case, changes in network voltage and frequency are considered sufficient for the test. Choice of rate of change of frequency variability attribute (up to ±1Hz/s) is based on values projected by the UK TSO. The test design follows a "Verification-Repeating" test design cluster (see Table 9.8).

Initial system state

For FuI stability verification tests:

- Inverter loading (3kW).
- Nominal network voltage and frequency (400V, 50Hz).

For islanding tests:

- PCC closed.
- Nominal network voltage and frequency (400V, 50Hz).

The inverters are loaded such that power flow across the PCC is up to 21% of inverter rating.

Evolution of system state and test signals

The test signals can be represented in a time series which constitutes the following:

- For FuI stability verification tests: the desired grid frequency ramp representing the RoCoF value of interest along with a constant load and DC source input values used for the initial system state.
- For FuI sensitivity verification tests: the desired load and inverter output values are selected to

maintain a PCC power flow of within the 1% of the inverter rating. A PCC change of state (i.e. open breaker) is initiated once this condition is satisfied after a few seconds time delay.

Other parameters

No other parameters are considered in this case.

Storage of data

The ‘measured parameters’ (as listed above) are recorded in a time series to perform the verification against the test criteria over the period of which the system state and test signals have evolved.

Temporal resolution

Real-time hardware testing and dynamic/transient events (e.g. changing of system frequency up to ±1Hz/s and opening of PCC to create and island) dictate a continuous temporal resolution from the test signal point of view.

Source of uncertainty

The following sources of uncertainty are considered when evolving the variability attributes:

- Power flows in the test setup: specifically the inverter output, the load consumption and the power flow through the PCC. This is influenced by the measurement equipment accuracy.
- Grid rate of change of frequency: this derived value is influenced by the measurement algorithm.

Suspension criteria / Stopping criteria

The suspension criteria in this example do not reflect an invalid test result, but are driven by the fact that the test is a ‘verification-repeating’ type of test where the FuI behaviour is observed in discrete time steps of up to 10min.

For FuI stability verification tests, the suspension criteria are the change of state of connection of the OuI or the completion of the test signal time series (i.e. frequency ramp).

For FuI sensitivity verification tests, the suspension criteria is the change of state of connection of the OuI or that a pre-defined time has elapsed if the OuI state of connection has not changed (in the order of 10min).

5.4 Experiment Specification

The experiment specification describes how a given test specification is mapped to a specific RI. Typically, there would be one experiment specification per test specification.

5.4.1 Motivation

An experiment specification builds on a given test specification and the specifics of a given lab infrastructure and provides the additional information required to carry out a concrete test or experiment in the lab. Thus, the realisation of a test at a specific RI is described in the experiment specification.

5.4.2 Methodology and Definitions

The experiment specification references the test specification it realizes.

Research Infrastructure: The name of the RI is given where the experiment is carried out.

Experiment realisation: The setup described in the test specification can be realised in different ways (e.g. simulation or involving hardware). Thus, a brief description of the realisation is given.

Experiment Setup: Here, a graphical and/or textual description is given that specifies the concrete lab equipment and its equipment. If simulators are used, give the corresponding tools and solvers. The experiment setup is a system configuration of SCType E-SC. In ERIGrid, it is based on RI database entries (RI-SC).

Experimental Design and justification: For all input parameters a reason is given why it has been chosen that way. This comprises:

- Concrete values or sequences of values of “variability attributes”,
- Concrete combinations of different variability attributes, and
- Number of repetitions for each combination.

Precision of equipment: For the components of the lab equipment the precision is given such that the experiment’s uncertainty can be derived.

Uncertainty of measurement: Based on the precision of lab instruments and of measurement algorithms, the parameters to model the measured quantities’ errors are provided.

5.4.3 Example

The following example is a continuation of 5.2.3. It exemplary shows the usage of the information introduced before.

Research Infrastructure

This experiment is specified for use in the Power Networks Demonstration Centre (PNDC) research infrastructure.

Experiment realisation

The experiment requires testing hardware PV inverters connected to a physical LV test grid or an LV grid simulator.

Using meta-modelling is not recommended at this stage due to the nature of the verification that is inherent to the OuI which is likely to vary in behaviour between different commercial offerings and regional settings. Once sufficient verification is achieved, meta-models can be developed and integrated to perform further verification or indeed characterisation (e.g. involving a large number of inverters) for a wide range of PoI’s.

Experiment Setup

The diagram in Figure 5.19 illustrates a high-level experiment setup that is in part specific to the PNDC laboratory. The lab equipment includes:

- Programmable DC power supplies with solar array emulation.
- Single and three phase PV inverters (3-10kW).
- Power quality meters.
- Source with controllable frequency. A motor generator set is used. The frequency set point including ramp rate is controlled via a scaled analogue control input to the source.
- Single and three phase LV distribution network. Normally an LV network diagram is required to establish the switching schedule but this depends on the laboratory. Changes in the network topology are also possible without affecting the test setup.
- An identified contactor/breaker is used as the PCC.

All interfaces are electrical:

- AC within the LV distribution network.
- DC input to the inverters: ±10Vdc to control source frequency.

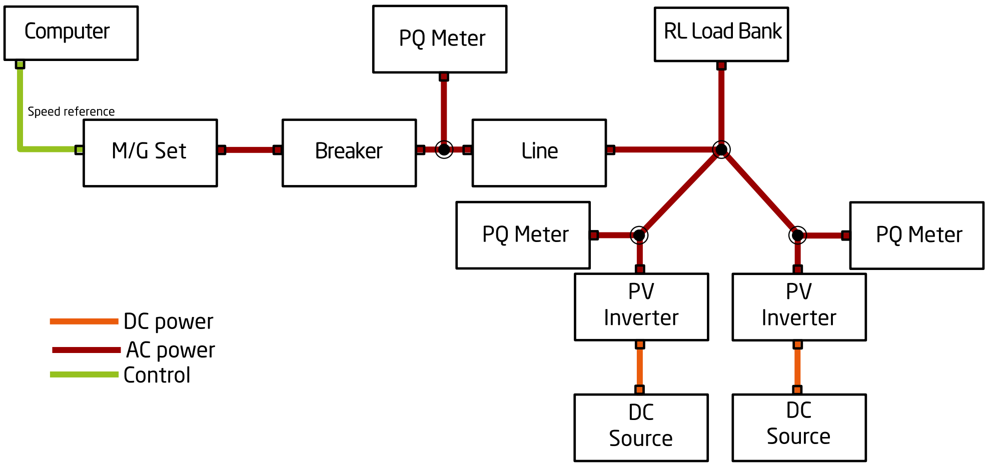


Figure 5.19: Realisation of test setup for inverter testing at RI

Experimental Design and justification

These are the main steps followed to conduct the experiment:

- Apply instrumentation to measure and record inverter terminal and PCC voltage and current
- Energise the AC network at nominal voltage and frequency.
- Switch on loads to achieve net power export from the grid source.
- Provide DC power to the PV inverters from the PV emulators and await inverter synchronisation to the grid.
- Verify initial conditions.
- Apply sensitivity or stability test signal
- Observe inverter response (trip/no trip).

Precision of equipment

Each verification test iteration is replicated three times.

Uncertainty measurement

Calibrated power quality meters are used to measure the electrical parameters defined in the test specification as ‘measured parameters’. The calibration ensures that the measurement equipment uncertainty is as specified by the manufacturer. The accuracy of the instrumentation used in this experiment was +/-0.1% and +/-0.5% for voltage and current measurements respectively.

6 Discussion and Conclusions

The methods outlined in this report define a heuristic framework for test specification, which has been aligned with state of the art specification methods and the vision for a “holistic testing” strategy.

Two major aspects have been introduced in this work are:

1. Definition of holistic testing concepts including the definition of test specification levels
2. A multi-domain method for system configuration description for each specification level

The holistic testing vision outlined in ERIGrid DoA bears a number of aspects which are widening the scope of conventional testing:

- Requirements associated with multiple domains are viewed as part of single test case
- Systematic and integrated testing strategy for systems, components and their integration
- The hybridization of methods applicable to distinct formal representation frameworks (i.e. ICT, discrete & logic oriented testing, vs. physical continuous models and uncertainty)
- The formal integration of several independent tests into a common framework
- Technical integration of different means of testing, such as real-time simulation

Addressing these requirements of a testing process, the proposed approach attempted to strike a balance between formal definitions, existing concepts within standards, and the practical use and understanding of tests. The practical use of the approach is supported by template forms with guiding text, graphical templates as well as exemplary applications.

The sequence of specifications includes:

1. **Test case** – in analogy to ‘use case’: defining the objectives and domains for a test
2. **Test specification** – what test is to be carried out? defining test system and its parameters
3. **Experiment specification** – how the test specification is to be implemented in a given RI

For each specification level a corresponding specification template and system configuration description method has been provided. Along with these the distinctions, three further types and contexts of system configuration specifications have been identified (see Section 3.2). We recognize that the wording of ‘tests’ vs. ‘experiments’ is not ideal. It has been carried over from the DoA and can be expected be replaced by a more suitable terminology in the future. Finally, the test case, test- and experiment specifications include the notions of test criteria and parameters which support the incremental scoping and definition of test factors for the application of the DoE method.

The preconditions for this work includes that there is a stark contrast between physical lab testing practices and ICT domain testing: the former are typically informal methods, associated with tacit and proprietary knowledge, and apart from the project partner contributions, no formal reference for specifications of physical testing; in contrast, there exist a number of formalized specifications in the ICT domain, both for system specification and test specification.

The notions and standards for testing are thus very domain specific, which motivated the foundational approach taken here. Nonetheless it was possible to build on prior efforts (e.g. in C-/P-HIL testing) and standards (UML/CIM, SGAM). As a result, the proposed system configuration description method is well aligned with current standards and feasible to implement for more practical test situations.

Further, as indicated in Section 2, there is a potential for further alignment and formalization of the method e.g. building directly on UML/SYSML, providing a rich toolset of formal techniques for e.g. model checking, model translation and verification. This however has been left for further work. The chosen simplification, in order to keep focus on necessary semantics.

7 Outlook

The results reported above set a foundation for future work within the ERIGrid project and beyond. There is a large number of avenues to continue the work, including both method improvements as well as complimentary work.

Direct improvements and follow-up work of the proposed approach have been suggested:

- Further formalization and toolset development to support its application, such as
  - formal specification and data extraction from system configuration diagrams
  - possibly the automatic generation of diagrams from a given data structure
- Evaluation of the complete approach on a realistic scale experiment
- Guidelines to the practical formulation of test criteria on the basis of standards requirements as well as on use case KPI
- A taxonomy of test criteria, along with guidelines for the corresponding formulation of test system and experiment configurations.
- A case library that could provide guidance to users of the approach in various testing situations

It is unclear whether all of these desirable developments can be achieved within the present project. However, more important to prove the feasibility of a complete holistic testing methodology will be the achievement of complementary developments, such as:

- Development of mapping concept, which will provide guidelines for application of this description methodology to the formulation of holistic test cases;
- The development of principles, guidelines and tools for the application of DoE methods;
- Concrete applications to jointly simulated and mixed hardware / software test;
- The demonstration of the overall method on a truly holistic test case

The ERIGrid project plan emphasizes these further developments and results can be expected within the project timeframe. There remain major methodological challenges in the optimal composition and quantitative evaluation of a holistic test case.

7.1 Outline of a Holistic Testing Architecture

ERIGrid's approach on Holistic Testing may be considered as a vision of a pre-standardised process and methodology implementing the testing of a system that includes multi-domain aspects (Power & ICT and eventually P-HIL). This vision can be extended to the mutualisation of resources of multiple partners to conduct parallel tests according to consortium-wide specifications and partner profiles.

We have outlined the main elements of a holistic testing procedure, in the framework of Smart Grid testing, following the ERIGrid project vision. In the process, some terminologies have been defined and a state of the art of holistic testing and DoE method in different technologies has been established. Establishing a standardised holistic testing procedure for applicable independent of research infrastructures is a long-term task because it requires strong interoperability among the participating research infrastructures. This implies however multiple agreements and harmonisations, and eventually changes of actual employed standards and protocols, toward a holistic procedure. Actual standardisation is beyond of the scope of ERIGrid.

In particular, several issues can be pointed out:

- Difference in actual infrastructures (protocol, standards, platform, etc.)
- Difficulty of partner profiling because of lack of information about which function in the considered infrastructure is necessary for the test.
- Difficulty in the adaptation of a test system to the partner infrastructure, due to a gap of the RI capabilities to the defined test procedure or resources.

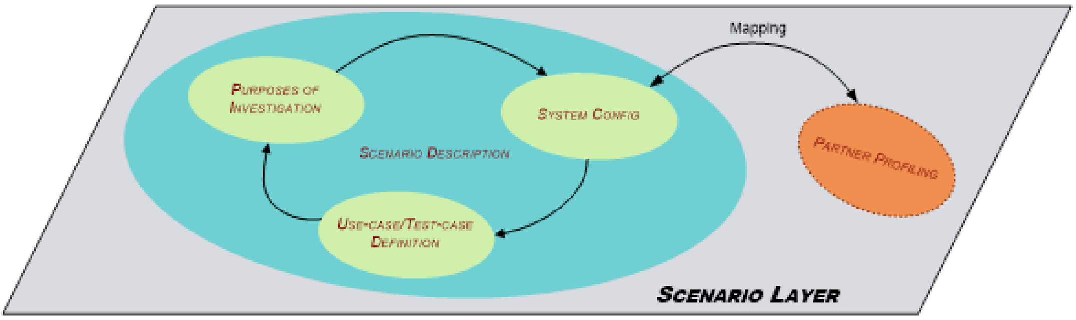
- Difficulty of exchange of test results and analysis because of difficulties in interoperability between data formats used in either RI.
- Difference in the scales of system considered at different RI may lead to different interpretation of the terminologies.
- Difficulty of implementing co-simulation or concurrent tests.

Inspired by the structure of the TTCN-3 testing description language (which was reviewed in Section 5.1.4.3), we outline below the conceptual structure of the required specifications in terms of layers. The standard TTCN-3 provides a complete and rational test suite for IT domain testing. Even though the standard is purely about testing in the ICT domain and has little relevance to the framework of ERIGrid project, the TTCN-3 testing process outline offers inspiration.

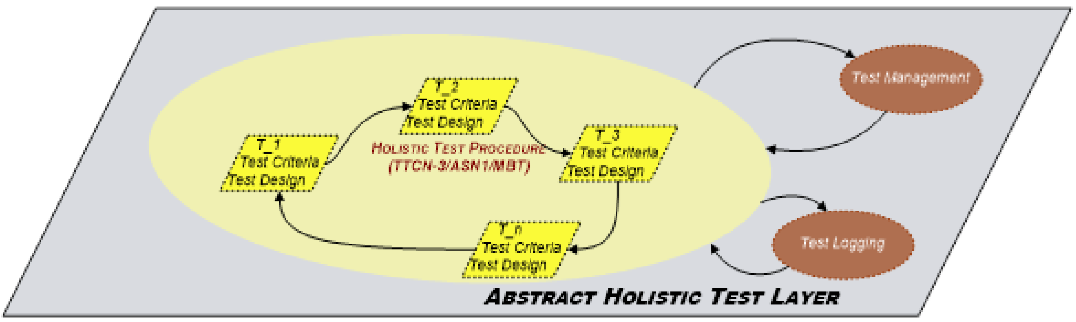
In general, we are aiming for *interoperability at information* and *application layers* via means of an *abstract test procedure.*

The standards and protocols mentioned should be considered as examples for the comprehensive purpose. They are, by no means, the only or recommended choice. The proposed holistic testing architecture consists of four layers:

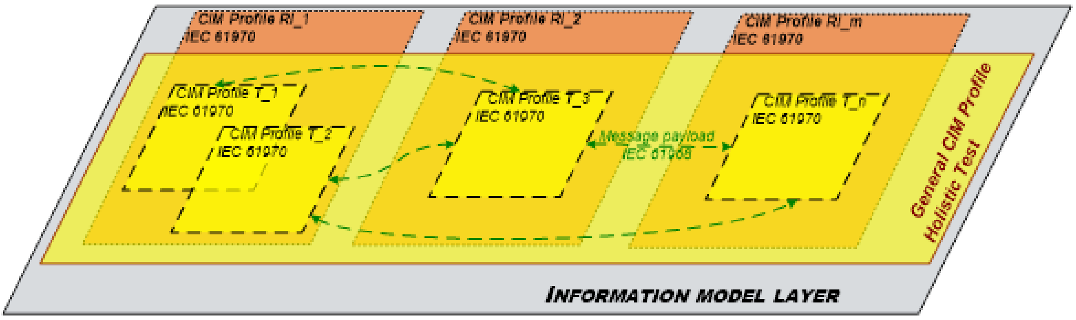
1. *Scenario layer:* In this layer, the chosen scenario is analyzed around the three main points: system configuration according to the scenario, purpose of investigation (deduced from the scenario and the desired research/contribution) and use-case/test-case definition. The partner profiling step in DoA defined procedure should fall into this layer, as the scenario and the system configuration influences strongly the experiments that a partner can offer.



2. *Holistic Testing layer*: this layer corresponds to the concept of abstract test suite of TTCN-3. The holistic testing procedure should be described in a specialized language (TTCN-3, MBT, ASN1, etc.). The dichotomies and communication among experiments will be defined and governed by the Test management and logging units.

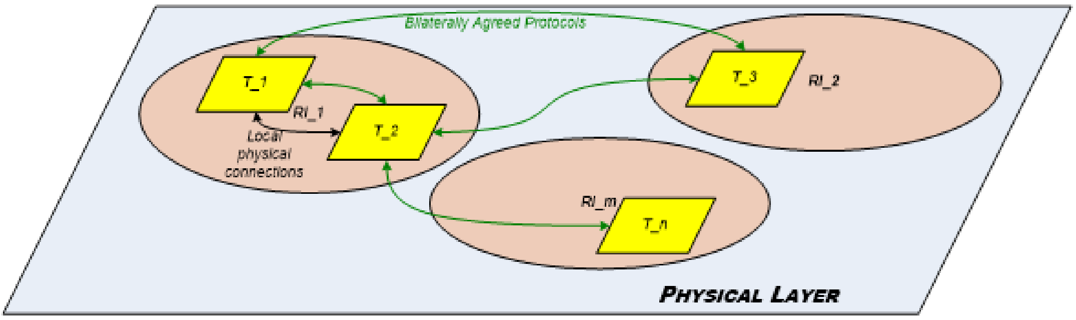


3. *Information model layer:* The information models of different tests are concretized into profiles and are mapped to the profile of different RI. The necessary message payloads model should also be defined.



4. *Physical layer:* the level in which the experiments are conducted and virtual exchange of information or physical intervention can take place. The profile of a test is mapped from the conceptual design via means of a platform adaptor and codec. For example: an experiment of controlling a generator A with algorithm B (written in TTCN-3). The model of generator A is extracted from the CIM library, while the algorithm B is adapted to the desired platform at local RI (C, C++, etc.). Judging by the number of TTCN-3 service providers, this step can be easily outsourced.

   The abstract profile is then adapted to the physical layer by means of different system adaptor. This step tolerates the eventual difference among local systems. For example, the test profile indicates syntax and necessary information for a communication. The actual communication can be done using available protocols at the local RI.

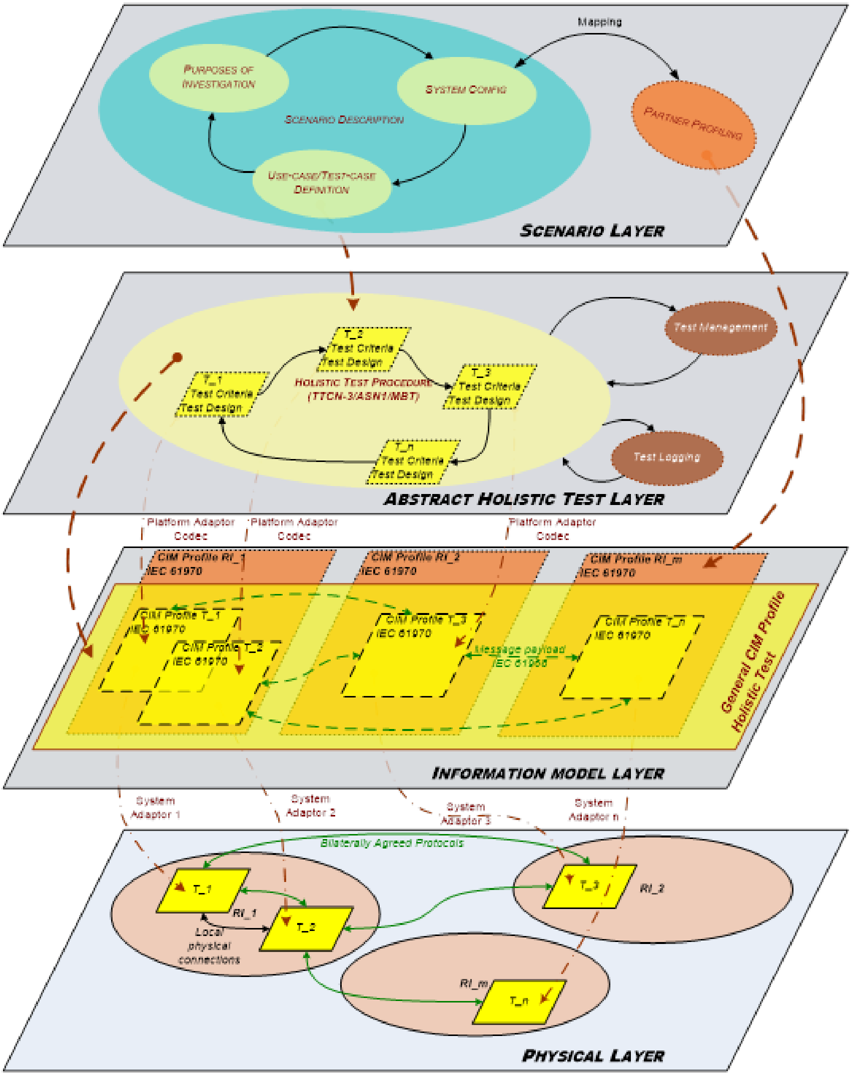


An overview of the full setup is provided in the following figure.

The proposed architecture maintains the general approach of holistic testing in the DoA and contributes by introducing some further advantages:

- Most of the work will be on the information model layer and holistic testing layer. No further alteration of existing experiments is needed.
- No effort is needed for harmonization and eventual adaptation of partner's available infrastructure.
- A single holistic testing procedure is maintained and followed. It leads to much less effort for test management and analysis.
- Testing and partner profiling follows a standardized and ontological approach, the outcomes are therefore extensible, adaptable and reusable. This property will be very valuable when an outside RI wants to integrate into the group.
- Possibility to fully automate the process via the test management system.

In general, this architecture allows interoperability at the information and application layers (cf. Section 4 - Use Cases - SGAM). Therefore, a persistent reference designation is provided with this schema. The aspect of 'persistent' KPI and test criteria refinement is not addressed in this outline, however. Defining such a strategy would require that the mapping concept and guidelines first have been developed.


Mapping
Purposes of Investigation
System Config
Scenario Description
Partner Profiling
Use-case/Test-case Definition
Scenario Layer
T_2 Test Criteria Test Design
Test Management
T_1 Test Criteria Test Design
Holistic Test Procedure (TTCN-3/ASN1/MBT)
T_3 Test Criteria Test Design
T_n Test Criteria Test Design
Test Logging
Abstract Holistic Test Layer
Platform Adaptor Codec
Platform Adaptor Codec
Platform Adaptor Codec
CIM Profile RI_1 IEC 61970
CIM Profile RI_2 IEC 61970
CIM Profile RI_m IEC 61970
CIM Profile T_1 IEC 61970
CIM Profile T_2 IEC 61970
CIM Profile T_3 IEC 61970
CIM Profile T_n IEC 61970
Message payload IEC 61968
General CIM Profile Holistic Test
Information Model Layer
System Adaptor 1
System Adaptor 2
System Adaptor 3
System Adaptor n
Bilaterally Agreed Protocols
T_1
RI_1
T_2
Local physical connections
T_3
RI_2
RI_m
T_n
Physical Layer

8 References

[1] F. C. Catalin, A. Miicea, V. Julija, M. Anna, F. Gianluca, and A. Elefherios, “Smart Grid Projects Outlook 2014,” European Commission - Joint research centre, JRC Science and Policy Reports, 2014.

[2] B. Galloway, G. P. Hancke, and others, “Introduction to industrial control networks.,” *IEEE Commun. Surv. Tutor.*, vol. 15, no. 2, pp. 860–880, 2013.

[3] V. H. Nguyen, Q. T. Tran, and Y. Besanger, “SCADA as a service approach for interoperability of micro-grid platforms,” *Sustain. Energy Grids Netw.*, vol. 8, pp. 26–36, Dec. 2016.

[4] I. P. W. Group, *IEEE Guide for Smart Grid Interoperability of Energy Technology and Information Technology Operation with the Electric Power System (EPS), End-Use Applications, and Loads*. 2011.

[5] T. S. G. I. Panel, “Introduction to NISTIR 7628: Guidelines for Smart Grid Cyber Security,” NIST Cyber Security Working Group, 2010.

[6] N. A. E. R. Corporation, “Security Guideline for the electricity sector: physical security,” NERC, 2011.

[7] P. Gallagher and G. Locke, “Guide for Applying the Risk Management Framework to Federal Information Systems - A security Life Cycle Approach,” Joint Task force Transformation Initiative, NIST Special Publication 800-37 Revision 1, Feb. 2010.

[8] I. E. Commision, “Power systems management and associated information exchange - Data and communications security - Part 12: Resilience and security recommendations for power systems with distributed energy resources (DER) cyber-physical systems,” IEC, IEC/TR 62351-12:2016, 2016.

[9] R. Brundlinger *et al.*, “Lab tests: Verifying that smart grid power converters are truly smart,” *IEEE Power Energy Mag.*, vol. 13, no. 2, pp. 30–42, 2015.

[10] T. Strasser *et al.*, “Towards holistic power distribution system validation and testing—an overview and discussion of different possibilities,” *E Elektrotechnik Informationstechnik*, pp. 1–7, 2016.

[11] CEN-CENELEC-ETSI Smart Grid Coordination Group, “Methodologies to facilitate Smart Grid system interoperability through standardization, system design and testing.” Oct-2014.

[12] T. Strasser *et al.*, “A Review of Architectures and Concepts for Intelligence in Future Electric Energy Systems,” *IEEE Trans. Ind. Electron.*, vol. 62, no. 4, pp. 2424–2438, Apr. 2015.

[13] M. Stifter *et al.*, “DG DemoNet validation: Voltage control from simulation to field test,” in *Innovative Smart Grid Technologies (ISGT Europe), 2011 2nd IEEE PES International Conference and Exhibition on*, 2011, pp. 1–8.

[14] W. W. Royce and others, “Managing the development of large software systems,” in *proceedings of IEEE WESCON*, 1970, vol. 26, pp. 1–9.

[15] A. Kossiakoff, W. N. Sweet, S. J. Seymour, and S. M. Biemer, *Systems engineering principles and practice*, vol. 83. John Wiley & Sons, 2011.

[16] D. Gelperin and B. Hatzel, “The growth of software testing,” *Commun. ACM*, vol. 31, no. 6, pp. 687–695, 1988.

[17] K. Beck, *Test Driven Development: By Example*, 1 edition. Boston: Addison-Wesley Professional, 2002.

[18] J. Kadubeck, “Holistic Testing Program Information,” *Holist. Test. Blog*, Apr. 2010.

[19] https://www. w3. or. w3c, “RDF Validator,” May 2016.

[20] Q. Group, “The Benefits of a Holistic Approach to Software Testing,” *Blog Qual.*

[21] CEN-CENELEC-ETSI Smart Grid Coordination Group, “Methodologies to facilitate Smart Grid system interoperability through standardization, system design and testing,” Oct. 2014.

[22] B. Prasad, “Sequential versus Concurrent Engineering—An Analogy,” *Concurr. Eng.*, vol. 3, no. 4, pp. 250–255, 1995.

[23] P. Gu and A. Kusiak, *Concurrent engineering: Methodology and applications*. Elsevier Science Inc., 1993.

[24] A. Kusiak, *Concurrent engineering: automation, tools, and techniques*. John Wiley & Sons, 1993.

[25] M. E. Paté-Cornell, “Uncertainties in risk analysis: Six levels of treatment,” *Reliab. Eng. Syst.*

Saf., vol. 54, no. 2textendash3, pp. 95–111, Nov. 1996.
[26] W. Lu and J. C. Doyle, “A State Space Approach to Robustness Analysis and Synthesis for Nonlinear Uncertain Systems,” May 1994.
[27] A. Fabrizi, C. Roos, and J. M. Biannic, “A detailed comparative analysis of x03BC; lower bound algorithms,” presented at the Control Conference (ECC), 2014 European, 2014, pp. 220–226.
[28] B.-W. Cheng and S. Maghsoodloo, “Optimization of mechanical assembly tolerances by incorporating Taguchi’s quality loss function,” *J. Manuf. Syst.*, vol. 14, no. 4, pp. 264–276, 1995.
[29] R. Y. Rubinstein and A. Shapiro, *Discrete event systems: Sensitivity analysis and stochastic optimization by the score function method*, vol. 346. Wiley New York, 1993.
[30] A. Jardin, W. Marquis-Favre, D. Thomasset, F. Guillemard, and F. Lorenz, “Study of a Sizing Methodology and a Modelica Code Generator for the Bond Graph Tool MS1,” presented at the Proceedings of the 6th International Modelica Conference, 2008, pp. 125–134.
[31] V. H. Nguyen, D. Eberard, W. Marquis-Favre, and L. Krahenbuhl, “Tolerance synthesis using bond graph inversion and fuzzy logic,” presented at the 2013 IEEE International Conference on Mechatronics (ICM), 2013, pp. 442–447.
[32] C. Scherer, P. Gahinet, and M. Chilali, “Multiobjective output-feedback control via LMI optimization,” *IEEE Trans. Autom. Control*, vol. 42, no. 7, pp. 896–911, 1997.
[33] “Wiley: Design and Analysis of Experiments, 8th Edition - Douglas C. Montgomery.” [Online]. Available: http://www.wiley.com/WileyCDA/WileyTitle/productCd-EHEP002024,subjectCd-ST23.html. [Accessed: 05-Mar-2017].
[34] “Wiley: Formal Methods for Industrial Critical Systems: A Survey of Applications - Stefania Gnesi, Tiziana Margaria.” [Online]. Available: http://www.wiley.com/WileyCDA/WileyTitle/productCd-0470876182.html. [Accessed: 05-Mar-2017].
[35] TC8, “Use case methodology - Part 2: Definition of the templates for use cases, actor list and requirements list (IEC 62559-2:2015),” Apr. 2015.
[36] IEC, “Electropedia - component,” *International Elekcrotechnical Dictionary*. International Electrotechnical Commission, 2014.
[37] IEC, “IEC 60050-351:2013 - International Electrotechnical Vocabulary - Part 351: Control technology,” Standard.
[38] J. Cabot and M. Gogolla, “Object constraint language (OCL): a definitive guide,” in *Formal methods for model-driven engineering*, Springer, 2012, pp. 58–90.
[39] EPRI Smart Grid Resource Center, “About the Use Case Repository.” .
[40] “Smart Grid Coordination Group reports.” [Online]. Available: https://bit.ly/1DFM68V.
[41] “EU FP7 Grid4EU project.” [Online]. Available: http://www.grid4eu.eu/.
[42] “EU FP7 Electra IRP project.” [Online]. Available: http://www.electrairp.eu/.
[43] “IEC 62559-2.” [Online]. Available: https://webstore.iec.ch/publication/22349.
[44] IEC, “Core IEC Standards.” [Online]. Available: http://www.iec.ch/smartgrid/standards/.
[45] NIST, “NIST Framework and Roadmap for Smart Grid Interoperability Standards, Release 3.0”,” 2014.
[46] KTH, “SGAM Template for EH2740.”
[47] “ERIGrid Deliverable D-JRA1.2, ‘Focal Use Case Collection,’” 2016.
[48] P. C. Kotsampopoulos, F. Lehfuss, G. F. Lauss, B. Bletterie, and N. D. Hatziargyriou, “The Limitations of Digital Simulation and the Advantages of PHIL Testing in Studying Distributed Generation Provision of Ancillary Services,” *IEEE Trans. Ind. Electron.*, vol. 62, no. 9, pp. 5502–5515, Sep. 2015.
[49] E. de Jong *et al.*, “European White Book on Real-Time Power Hardware in the Loop Testing : DERlab Report No. R- 005.0,” DERlab e.V. – European Distributed Energy Resources Laboratories, Report, Mar. 2012.
[50] I. F. Abdulhadi *et al.*, *International White Book on DER Protection: Review and Testing Procedures*. DERlab eV–European Distributed Energy Resources Laboratories, 2011.
[51] “IEEE Guide for Power System Protection Testing,” *IEEE Std C37233-2009*, pp. 1–112, Dec. 2009.

[52] A. Grid, “Network protection & automation guide,” *Alstom Grid*, 2011.
[53] M. Achterkamp, R. Cremers, A. Valdivielso, E. Navarro, and F. D. O. Carvalho, “Power system protection relay testing moving from certification testing to application testing,” 2004.
[54] P. Forsyth, T. Maguire, and R. Kuffel, “Real time digital simulation for control and protection system testing,” in *Power Electronics Specialists Conference, 2004. PESC 04. 2004 IEEE 35th Annual*, 2004, vol. 1, pp. 329–335.
[55] C. Dufour, “A Combined State-Space Nodal Method for the Simulation of Power System Transients,” *Ieee Power Energy Soc. Gen. Meet.*, p. 1 pp., 2011.
[56] P. Kotsampopoulos, N. Hatziargyriou, B. Bletterie, G. Lauss, and T. Strasser, “Introduction of advanced testing procedures including PHIL for DG providing ancillary services,” in *IECON 2013 - 39th Annual Conference of the IEEE Industrial Electronics Society*, 2013, pp. 5398–5404.
[57] M. Pol, T. Koomen, and A. Spillner, “Management und Optimierung des Testprozesses,” *Heidelb. Dpunkt Verl.*, 2000.
[58] I. C. Society, P. Bourque, and R. E. Fairley, *Guide to the Software Engineering Body of Knowledge (SWEBOK(R)): Version 3.0*, 3rd ed. Los Alamitos, CA, USA: IEEE Computer Society Press, 2014.
[59] J. Tian, *Software Quality Engineering: Testing, Quality Assurance, and Quantifiable Improvement*. John Wiley & Sons, 2005.
[60] “TTCN3 Tutorial.” ETSI Center for Testing and Interoperability.
[61] T. Hwang, Y. Yoo, H. Lee, and I. Lee, “Design of Automated Smart Grid CIM Interoperability Test System Based on TTCN,” in *International Conference on Informatics & Applications*, 2014.
[62] D. E. M. Bondy, O. Gehrke, A. Thavlov, K. Heussen, A. M. Kosek, and H. W. Bindner, “Procedure for validation of aggregators providing demand response,” in *2016 Power Systems Computation Conference (PSCC)*, 2016, pp. 1–7.
[63] K. Heussen, M. Uslar, and C. Tornelli, “A use case methodology to handle conflicting controller requirements for future power systems,” in *Smart Electric Distribution Systems and Technologies (EDST), 2015 International Symposium on*, 2015, pp. 582–587.

9 Annex

9.1 List of Figures

9.2 List of Tables

9.3 System Configuration Appendix

The system configuration description can take several forms. The graphical representation was discussed in Section 3.2, where a tabular description is also presented. In this appendix we present a more thorough example of the tabular description method.

9.3.1 Domains and Components

For the use case we can identify the following domains:

- Electrical power system
- ICT
- Control system
- Stakeholders

The systems and components are listed as follows.

Table 9.1: Component table

System	Sub-system	Component	Attributes
Distribution network (MV/LV)	Lines	Cables	Electrical parameters Physical parameters Economic parameters
	Transformers	Tap changer	Voltage levels Electrical values Losses Tap changer steps Controller type (OLTC)
	DER units	Energy source Generator Inverter External communication interface	Power range Power factor Voltage level Electrical values Controllability Ramp rates Communication protocols
	Storage units	Storage unit Inverter Management system (BMS etc.)	Capacity Peak power charge/discharge SOC status Voltage level Controllability Ramp rates Communication protocols
	IEDs (Intelligent Electronic Devices)	Controllers Measurement units	Interface and protocol Functionalities Data model
Central control	Coordinated voltage control	Central Controller Tap changers Compensation units Reactive power control Storage units Generator control	Power rates Power factor limits Ramp rates Control capability Voltage limits Power losses Tap changer step
Metering system	Smart meters	Measurement unit Communication unit	Meter type Measuring capabilities (power quality, etc.) Communication channel and protocol

Measurement system	Phasor measurement units (PMUs)	Voltage sensors Communications network Time synchronisation (e.g. by GPS)	Sensor types PMU configuration (list and format of measurements collected by each PMU) PMU accuracy (from calibration testing) PMU locations Communications protocols
Communication systems	Communication links	Physical communication media Routers	Bandwidths Delays Jitter Errors Package losses Protocols Information models Redundancy Time synchronisation Security and encryption
Data management	Management systems	DMS	Interfaces Modularity / interoperability Protocols Information models
Actors	System operators	DSOs	

9.3.2 Connectivity

In a generic system configuration, the connectivity is focused on the types of relevant connections as well as domains. It can be represented graphically, as illustrated in Section 3.3, or using a table form. A connectivity matrix illustrates the connectivity types between different component types, referencing the abstract connectivity types listed in Table 3.9, as illustrated in Table 9:2, below. This illustration of connectivity has been developed in the form of a matrix in which domains and components are listed and their connectivity is mapped using the abstract connectivity types. Such a matrix can serve to map both intra-domain and inter-domain connectivity. Another, more concise annotation relates the components to domains, and thus indicates the allocation of component terminals to domains. The example provided in Table 9.3 is loosely based on the generic system configuration "Vertical Integration" presented in D-JRA1.1.

For specific connectivity, the graphical representations or a topology table based on terminals and connectivity nodes is appropriate, as introduced in Section 3.2.3.3.

Table 9:2: Connectivity matrix

	Electrical power system	-Distribution network	--Lines	--Transformers	--DER units	--Storage units	--IEDs	**Control system**	-Central Control	--Coordinated Voltage Control	**ICT**	-Metering Systems	-Measurement Systems	-Communication systems	-Data management	Stakeholders	-DSO
Electrical power system																	
-Distribution network																	
--Lines				DP	IP	IP	DD			DD		DP			AD		DD
--Transformers			DP		IP	IP	DD			DD		DD			DD		DD

	Electrical power system	-Distribution network	--Lines	--Transformers	--DER units	--Storage units	--IEDs	**Control system**	-Central Control	--Coordinated Voltage Control	**ICT**	-Metering Systems	-Measurement Systems	-Communication systems	-Data management	Stakeholders	-DSO
--DER units			IP	IP			DD			DD		DD	DD	DD	DD		DD
--Storage units			IP	IP			DD			DD		DD	DD	DD	DD		DD
--IEDs			DD	DD	DD	DD				DD		DD	DP	DD	AD		DD/AD
Control system																	
-Central Control																	
--Coordinated Voltage control			DD	DD	DD	DD	DD					DD	DD	ICC	AD		DD
ICT																	
-Metering Systems			DP	DD	DD	DD	DD			DD				ICC	AD		DD
-Measurement Systems					DD	DD	DP			DD				DD	AD		DD
-Communica-tion systems					DD	DD	DD			ICC		ICC	DD		DD		DD
-Data management			AD	DD	DD	DD	AD			AD		AD	AD	DD			DD/AD
Stakeholders																	
-DSO			DD	DD	DD	DD	DD/AD			DD		DD	DD	ICC	DD/AD		

Table 9.3: Connectivity table – Component/Domain form for Domain identification

Component	Electricity – Trans-mission	Electricity - Distribution	Electricity - Conversion	ICT - Market & Enterprise	ICT – Operations	Stake-holders	Heat Distribution	Transport (local)	Energy Ressource or Consumer
ZIP-Loads (Equivalent-Model)	DP		DP						DP
Controllable loads (P/Q)	DP		DP						DP
PV park	IP		DP						DP
Connected Grids	DP								
Transmission Substation	DP	DP							
Switches/Breaker	DP								
Batteries	IP		DP						
Microgrid (PCC-equivalent)		IP	DP						DP
Small hydro (flowing water/in stream)		IP	DP						DP
Distribution Air lines	DP								
Distribution Cables	DP								

Component	Electricity – Trans-mission	Electricity - Distribution	Electricity - Conversion	ICT - Market & Enterprise	ICT – Operations	Stake-holders	Heat Distribution	Transport (local)	Energy Ressource or Consumer
Distribution Substation	DP				DD				
Transformer (OLTC)	DP	DP							
Trading application				DD		AD			AD
Market database and clearing system	AD			DD		AD			
Business Operation Systems	AD			DD, AD	AD				AD
Customer Relationship Management Systems		AD		AD	AD	AD			
AMI Database, MDM Software		AD				AD			
External IT applications									
EMS (Energy Management Systems) {TSO,BRP}					DD, AD				
DMS (Distribution Management Systems)					DD, AD				
Engineering and maintenance applications	AD	AD					AD		
SCADA (incl. State Estimator, OPF, …)	DD	DD			DD				
Mapping services (e.g. proto-col mapping)				AD	AD				
Data Concentrator		AD			DD				
Control devices (e.g. power con-verters, OLTCs, switch-gear (relays, breakers,), IEDs)		IP, DD			DD				
Substation / DER / Hydro devices		IP, DD			IP, DD				
Control Centers									
Measurement devices (e.g. PMU, RTU, Smart Meters)									
Home Area Networks				ICC	ICC				
Neighbourhood Area Network				ICC	ICC				
Wide Area Networks				ICC	ICC				
DSO's		R,D,O,OP,I							
Heat Pump		DP							DP
CHP		DP							DP
Electric Heating		DP							DP
Light bulb		DP							DP
Heat storage		IP							DP

Component	Electricity – Trans-mission	Electricity - Distribution	Electricity - Conversion	ICT - Market & Enterprise	ICT – Operations	Stake-holders	Heat Distribution	Transport (local)	Energy Ressource or Consumer
Battery Electric Vehicle (BEV)		IP						DP	
Plug-in Hybrid Electric Vehicle (PHEV)		IP						DP	
Charging post (smart meter)		IP, DD			DD				
Smart Charger		IP			DD				
Charging Station		DP			DD, AD			DP, AD	

9.3.3 Component Attributes

The following table illustrates the relevant component attributes for the Generic System Configuration introduced in Section 3.3.1, the system configuration is illustrated Figure 3.10: Graphical representation of the TC-GSC for the coordinated voltage control test case

Table 9.4: Attribute table

Type of component	Attributes	Additional info
Electricity related	Energy cost	€/MWh
	Capacity cost	€/MW
	Voltage	V
	Current	A
	Active Power	W
	Reactive power	VAr
	Power rating	VA
	Nominal Voltage	V
	Cost	€
Primary sources / generation related	Renewable / Non-renewable	
	Conventional / Non-conventional	
	Intermittency / Controllability	
	Availability	
	Onshore	solar
	Cost	
Generation / consumption related	Voltage levels	Depending on the country. HV/MV/LV, Transmission/Distribution
	Power/Sizes	micro<100kW, mini<1MW, small<10MW, medium<100MW, big>100MW
	Power range	PV panels, BESS

Type of component	Attributes	Additional info
	Cost	Energy production, Capacity, infrastructure, IRR, amortization, operational costs, incentives
	Ramping performance	on-peak, base energy
	Efficiency	
	Reliability	
	Onshore	Power plants, PV panels…
Stakeholder related	Ownership of stakeholder	Public / private / public-private joint venture
	Generation capacity	
	Consumer type	
	Regulator authority type	Country specific standards/ compliance with European norms/ compliance with international norms
Global Attributes	Temperature	
	Weather related data	solar radiation profiles

9.4 Use Case Definition Example

9.4.1 Use Case Description

In the following example, the function “Optimal centralized coordinated voltage control” is described, along with its mapping on SGAM.

The use case describes how this kind of function can adjust system settings in case of deep impact of Distributed Energy Resources on a monitored network.

The template employed here is based on the IEC PAS 62559 Use Case template, however, based on an adaptation developed in the ELECTRA project, which introduces additional fields and specific field interpretations to render the template more suitable for the specification of control functions [70].

1 Description of the use case

1.1 Name of use case

Use case identification		
ID	***Area Domain(s)/ Zone(s)***	***Name of use case***
ICCS_NTUA_ex.1	Zones: ?? Domains: Electric Power, control	Optimal centralized coordinated voltage control

1.2 Version management

Version management				
Version No.	***Date***	***Name of author(s)***	***Changes***	***Approval status***
1.0	06-06-2016	Marios Maniatopoulos, Panos Kotsampopoulos	initial setup of the use case	Draft, Work in Progress

1.3 Scope and objectives of use case

Scope and objectives of use case	
Scope	Optimally control the voltages of a distribution network, while simultaneously minimizing power losses and tap change operations of the transformer's on-load tap changer (OLTC). This is accomplished with a central controller that receives real-time measurements from key nodes of the network, solves an optimization problem, and dispatches set-points to controllable devices located in the network, such as the OLTC, inverters of DER units and storage systems.
Objective(s)	O1: Minimize voltage deviations from the nominal value O2: Minimize power losses O3: Minimize tap change operations of the OLTC
Related Higher-level use case(s)	Voltage Control
Control Domain Reference	Distribution network

1.4 Narrative of use case

Narrative of use case
Short description
A central controller receives real-time measurements from key nodes of the distribution grid via a communication network and solves an optimi-

zation problem with the aim to minimize voltage deviation from the nominal value, power losses, and tap change operations of the transformer's OLTC. The outputs of the optimization are set-points that are transmitted via the communication network to flexible devices located in the distribution grid, such as the OLTC, inverters of DER units and storage systems.
Complete description
A central controller is installed at substation level and is initialized with all the necessary static data of the network: network topology, admittance of lines and transformer, nominal power of DER units and storage systems, operating limits of DER units, storage systems and OLTC. While it operates, it requests and receives real-time power measurements from the smart meters of loads and DER units, as well as the state of charge (SOC) of the storage systems and the current tap position of the OLTC, in discrete iterations (e.g. every 10 minutes). Using this dynamic data, it formulates an optimal power flow problem, whose objective function involves the minimization of voltage deviation of critical nodes from the nominal value, power losses of the lines and transformer, and tap change operations of the OLTC. The outputs that result from the solution of this optimization are set-points for all controllable devices located in the network. Specifically, the controller calculates a set-point for the tap position of the OLTC, reactive power set-points for the inverters of DER units, as well as active and reactive power set-points for the storage systems. The reception of measurements and transmission of set-points is carried out through a communication network.

1.5 Optimality Criteria

(Directly associated with objectives. E.g. by what metric to 'minimise' something)

Optimality Criteria			
ID	***Name***	***Description***	***Reference to mentioned use case objectives***
1	Voltage deviation	Maintain critical node voltages close to the nominal	O1
2	Power losses	Minimize power losses on the lines and transformer	O2
3	Tap change operations	Minimize the number of tap change operations of the OLTC	O3

1.6 Use case conditions

Use case conditions
Assumptions
The central controlled is assumed to be located at substation level and it is operated by the controlled network's DSO. The controller has direct communication with the DMS (Distribution Management System), through which it receives and transmits data.
Prerequisites
- Smart meters must be installed at every load and DER unit - Smart meters must be able to transmit power data in real-time - A communication network must exist between the DMS, the Smart Meters and the local controller of each device.

-The inverters of the DER units and storage systems must be controllable and able to operate at power factors <1 (both leading and lagging)

1.7 Further information to the use case for classification / mapping

Classification Information
Relation to other use cases
Level of depth
Prioritisation
Generic, regional or national relation
Control Mechanism (from Taxonomy)
Centralized
Further keywords for classification
Voltage Control, Centralized Control, Coordinated Control

1.8 General remarks

General remarks
The central controller in this use case receives dynamic data through a communication network, solves an optimal power flow problem (minimization of an objective function), and dispatches set-points to all controllable devices of the distribution network.

2 Diagrams of use case

Diagram(s) of use case
See next Section “SGAM Mapping”

3 Technical details

3.1 Actors

Actors			
Grouping		***Group description***	
Operation actor		Actors that take part in this use case	
Actor name	***Actor type***	***Actor description***	***Further information specific to this use case***
DSCC	Central controller	Central controller	The central controller responsible for solving the optimal power flow and dispatching the controllable de-

			vices of the network
DMS	Distribution Management System	The management system of the network's DSO	
DERGen	DER units	Distributed energy resource units (PVs, Wind Turbines, etc.)	
DERStorage	Storage systems	Storage systems (batteries, flywheels, etc.)	
DERGen Controller	Local controller	Control of DER units	A local controlling device for each DER unit responsible for receiving the reactive power set-point from the central controller and applying it
DERStorage Controller	Local controller	Control of Storage systems	A local controlling device for each storage system responsible for receiving the active and reactive power set-points from the central controller and applying them
OLTC	On-Load Tap Changer	The On-Load Tap Changer mechanism of the substation's transformer	
OLTC Controller	Local controller	Control of OLTC	A local controlling device for the OLTC responsible for receiving the tap position set-point from the central controller and applying it
Smart Meter	Smart Meter	A smart meter installed at each load and DER unit that is capable of transmitting power data	
Data Concentrator	Data concentrator	Device responsible for concentrating Smart Meter measurements	

3.2 References

References						
No.	**References type**	**Reference**	**Status**	**Impact on use case**	**Originator / organisation**	**Link**

4 Step by step analysis of use case

4.1 Overview of scenarios

Scenario conditions						
No.	**Scenario name**	**Scenario description**	**Primary actor**	**Triggering event**	**Pre-condition**	**Post-condition**
1	Controller Initialization	The controller is initialized with all the necessary static data of the network that it will control	DSCC	Controller installation/Network infrastructure update	-Controller installed and running -Communications established	-The controller has acquired all necessary static data
2	Measuring	Real-time power measurements from Smart Meters are transmitted to the DSCC, as well as the state of charge of storage systems and the tap position of the OLTC	Smart Meter	Periodical (discrete iterations)	-Controller initialized with the network's static data -Communications established -OLTC Controller up and running -DERStorage Controller up and running	The central controller has received power measurements from loads and DER units, the state of charge of storage systems and the tap position of the OLTC
3	Optimal power flow	The central controller uses the received dynamic data to formulate and solve the optimal power flow problem	DSCC	Reception of dynamic data complete	-Controller up and running -Reception of dynamic data complete	The controller has solved the optimal power flow problem and has calculated the set-points for the controllable devices of the network
4	Set-point dispatch	The central controller transmits the calculated optimal set-points to all the controllable devices of the network	DSCC	Optimal power flow solution achieved	-Controller up and running -Communications established -OLTC Controller up and running -DERStorage Controller up and running -DERGen Controller up	All controllable devices of the network have received and applied the set-points sent by the central controller

					and running	

4.2 Steps – Scenarios

Regarding the “Service” column, the possible values are consistent with those provided by IEC 61969-100 (section 6.2.2). In particular the following verbs can be used:

- GET: used to query for objects
- CREATE: used to create objects
- DELETE: used to delete objects
- CLOSE and CANCEL: imply actions related to business processes, such as the closure of a work order or the cancellation of a control request
- CHANGE: used to modify objects
- EXECUTE: is used when a complex transaction is being conveyed, which potentially contains more than one verb

The response to each of the above requests uses the ‘reply’ verb. Also, events may be generated by using the verbs CREATED, DELETED, CLOSED, CANCELED, CHANGED and EXECUTED.

Scenario								
Scenario name :		Controller Initialization						
Step No.	***Event***	***Name of process/ activity***	***Description of process/ activity***	***Service***	***Information producer (actor)***	***Information receiver (actor)***	***Information exchanged (IDs)***	***Requirements R-ID***
1	Initialization	Request topology matrix	DSCC requests the distribution system topology matrix from DMS		DSCC	DMS	1	
2	Request received	Reply topology matrix	DMS transmits the topology matrix of the distribution system to DSCC		DMS	DSCC	5	
3	Topology Matrix received	Admittance matrix formation	DSCC forms the admittance matrix of the distribution system		DSCC	DSCC		
4	Initialization	Request system parameters	DSCC requests all constant system parameters from		DSCC	DERGen Controllers, DER-Storage Con-	1	

			DMS, DERGen, DERStorage and OLTC Controllers			troller, OLTC Controller		
5	Request received	Reply system parameters	DMS transmits the maximum and minimum allowable voltage limits of the distribution system		DMS	DSCC	8	
6	Request received	Reply system parameters	DERGen Controllers transmit the minimum allowable power factor and nominal apparent power of each DER unit to DSCC		DERGen Controllers	DSCC	8	
7	Request received	Reply system parameters	DERGen Controllers transmit the minimum allowable power factor and nominal apparent power of each DER unit to DSCC		DERStorage Controller	DSCC	8	
8	Request received	Reply system parameters	OLTC Controller transmits the total number of available tap positions and the step value to DSCC		OLTC Controller	DSCC	8	
Scenario								
Scenario name :		Measuring						
Step No.	***Event***	***Name of process/ activity***	***Description of process/ activity***	***Service***	***Information producer (actor)***	***Information receiver (actor)***	***Information exchanged (IDs)***	***Requirements R-ID***
9	Periodical	Active power measurement	Smart meter acquires active power		DERGen	Smart Meter	2	

			measurements					
10	Periodical	Transmission of measurement	Smart meter transmits active power measurements		Smart Meter	Data Concentrator	2	
11	Periodical	Transmission of measurement	Data concentrator transmits active power measurements		Data Concentrator	DMS	2	
12	Periodical	Active and reactive power measurements	Smart meter acquires active and reactive power measurements		Loads	Smart Meter	2	
13	Periodical	Transmission of measurements	Smart meter transmits active and reactive power measurements		Smart Meter	Data Concentrator	2	
14	Periodical	Transmission of measurements	Data concentrator transmits active and reactive power measurements		Data Concentrator	DMS	2	
15	Periodical	Transmission of measurements	DMS collects all real-time data received from the Smart Meters and transmits it to DCSS		DMS	DSCC	2	
16	Periodical	Request state of charge	The DSCC requests the state of charge of the storage systems		DSCC	DERStorage Controller	1	
17	Periodical	Reply state of charge	The DERStorage Controller transmits the state of charge of the storage system		DERStorage Controller	DSCC	6	
18	Periodical	Request current tap position	The DSCC requests the current tap position of the OLTC		DSCC	OLTC Controller	1	

19	Periodical	Reply current tap position	The OLTC Controller transmits the current tap position of the OLTC		OLTC Controller	DSCC	7	

Scenario								
Scenario name :	Optimal power flow							
Step No.	**Event**	**Name of process/ activity**	**Description of process/ activity**	**Service**	**Information producer (actor)**	**Information receiver (actor)**	**Information exchanged (IDs)**	**Requirements R-ID**
20	Measurements received	Optimal power flow	The DSCC formulates and solves the optimal power flow problem		DSCC	DSCC		

Scenario								
Scenario name :	Set-point dispatch							
Step No.	**Event**	**Name of process/ activity**	**Description of process/ activity**	**Service**	**Information producer (actor)**	**Information receiver (actor)**	**Information exchanged (IDs)**	**Requirements R-ID**
21	Optimal power flow solution achieved	Reactive power set-point dispatch	The DSCC transmits the reactive power set-points to the DERGen Controllers		DSCC	DERGen Controllers	3	
22	Set-point received	Application of the reactive power set-point	The DERGen Controllers update the operating parameters of the DER units by applying the received set-points		DERGen Controllers	DERGen	4	
23	Optimal power flow solution achieved	Active and reactive power set-points dispatch	The DSCC transmits the active and reactive power set-points to the DERStorage Controllers		DSCC	DERStorage Controllers	3	
24	Set-point received	Application of the active and reactive power	The DERStorage Controllers update the operating param-		DERStorage Controllers	DERStorage	4	

		set-points	eters of the storage systems by applying the received set-points					
25	Optimal power flow solution achieved	Tap position set-point dispatch	The DSCC transmits the tap position set-point to the OLTC Controller		DSCC	OLTC Controller	3	
26	Set-point received	Application of the tap position set-point	The OLTC Controller updates the tap position of the OLTC by applying the received set-point		OLTC Controller	OLTC	4	

5 Information exchanged

Information Exchanged			
Information exchanged ID	***Name of information exchanged***	***Description of information exchanged***	***Requirements IDs***
1	Request	Request for data	
2	Power measurement	Measurement of active and/or reactive power	
3	Set-point	Set-point transmitted by the DSCC to the controllable devices	
4	Operating parameters	Updated operating parameters of a component	
5	Static data	Static data of the network	
6	State of charge	The state of charge of the storage systems	
7	Tap position	The current tap position of the oltc	
8	Component nominal operating limits	The nominal operating limits (e.g. nominal power, minimum power factor, etc.) of all components of the network	

6 Requirements

Requirements		
Categories ID	***Category name for requirements***	***Category description***

Requirement ID	Requirement name	Requirement description

7 Common Terms and Definitions

Common terms and definitions	
Term	**Definition**

8 Custom information (optional)

Custom information (optional)		
Key	**Value**	**Refers to section**

9 Controller Conflicts and Misuse cases

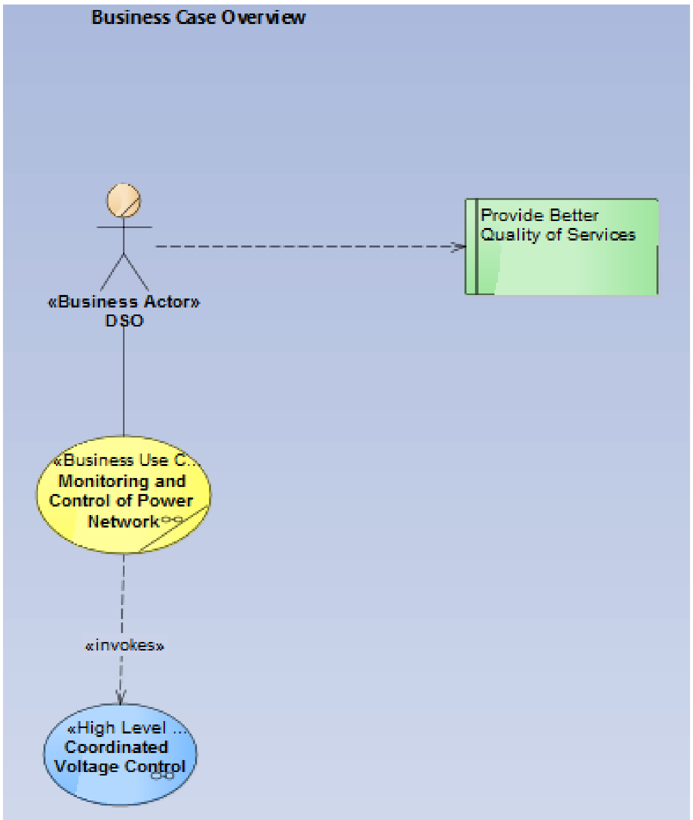
A controller conflict is an undesired change of an intended control action as a response to another control action.

Controller conflict cases								
id	**Case Name**	**Description**	**Conflict References**	**Related Actors**	**Related Use Cases**	**Related Requirements**	**Related Objective**	**Recommended mitigation**
<line_text>	<line_text>	<text>	<text>	Ref.	Ref.	Ref.	Ref.	<text>

9.4.2 SGAM Modeling

Business Layer

Coordinated voltage control is an important functionality that contributes in monitoring and control of the power network. Towards this direction, DSOs can provide better quality of services.

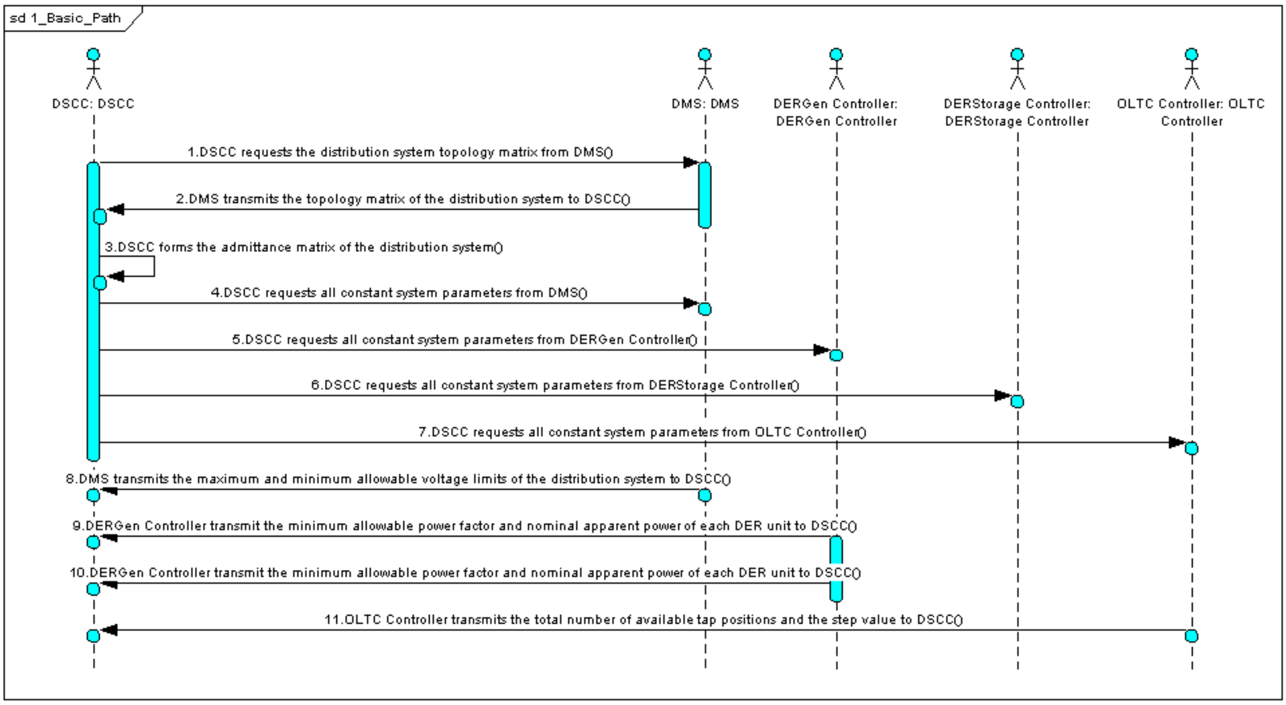


SGAM business layer

Function Layer

Controller Initialization

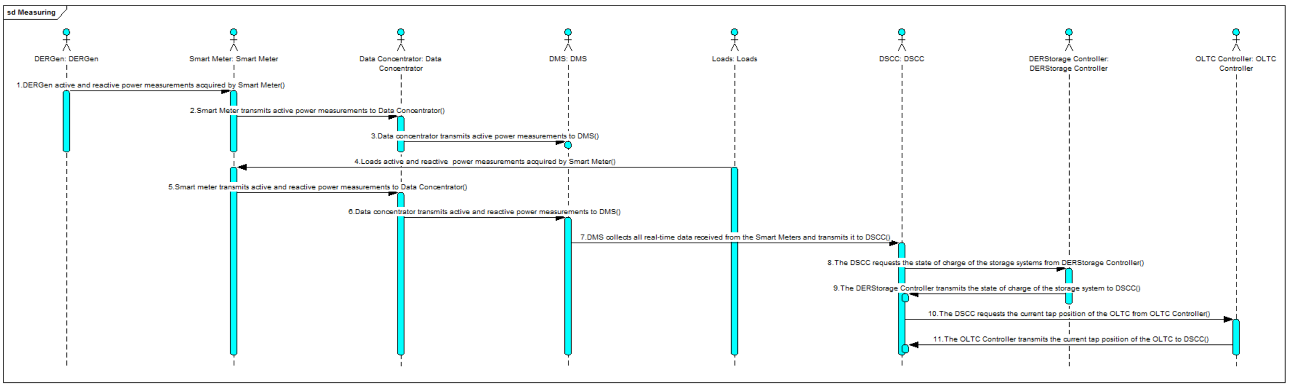
This scenario describes all the appropriate steps that are required to initialize the controller. In order for the DSCC to be totally ready for the control procedure, it should be aware of the current state of the system. In this procedure the DSCC request all the essential information of the network that it will control. Network topology, admittance of lines and transformer, nominal power of DER units and storage systems, operating limits of DER units, storage systems and OLTC are the required information for the initialization of the DSCC. DMS is responsible for transmitting to the DSCC the topology of the grid as well the maximum and minimum allowable voltage limits of the distribution system. In addition, DSCC requests directly from the DER/Storage/Load Controllers the minimum allowable power factor and nominal apparent power of each DER/Storage/Load unit. Finally, DSCC should also be informed about the total number of available tap positions and the respective step value of OLTC.



Controller Initialization

Measuring

This scenario describes the measuring procedure (data acquisition) that is necessary for the operation of the central controller. Here, smart meters acquire power measurements from the DER/Storage units and loads, which they transmit to a data concentrator. Subsequently, the data concentrator sends this data to the DMS in predefined time intervals, which, in turn, transmits it to the DSCC. At the same time, the DSCC requests real-time information about the SoC of the storage systems and the current tap position of the OLTC directly from their respective controllers.



Measuring

Optimal Power Flow

Upon the acquisition of the real-time data of the grid components the DSCC formulates and solves the optimal power flow problem.

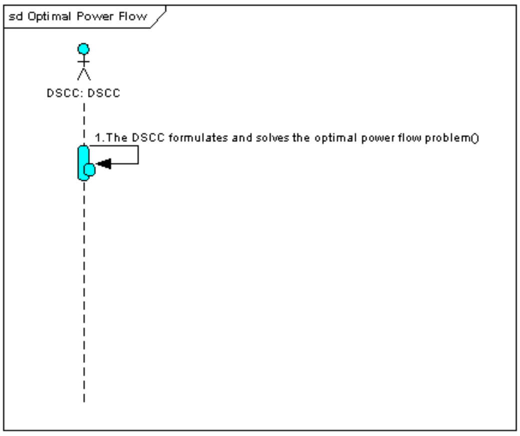
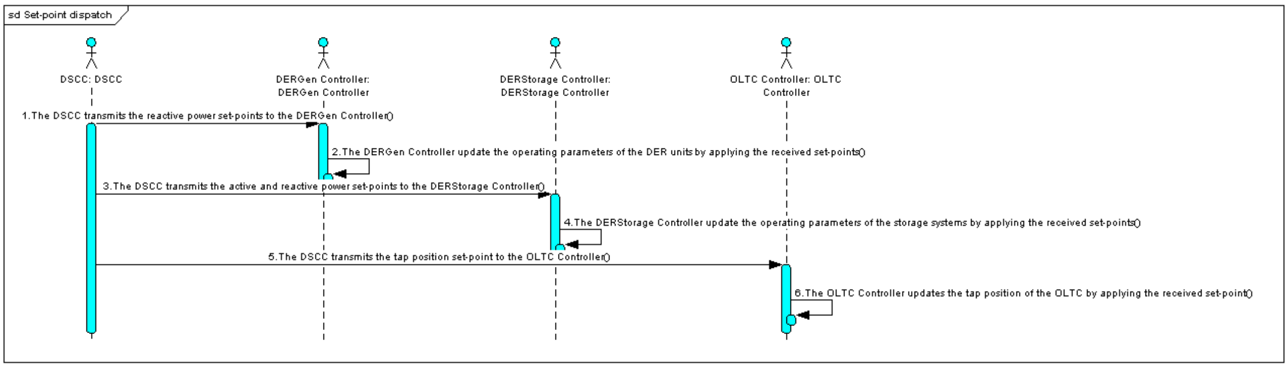


Figure: Optimal Power Flow

Setpoint Dispatch

Upon the solution of the Optimal Power Flow problem, the dispatch of the calculated setpoint for the various voltage controlling components of the distribution network follows. Firstly, the DSCC transmits the optimal setpoint for the DER/Storage units and the new tap position of the OLTC to their respective controllers. Then, the controller of each component applies the acquired optimal operating setpoint.

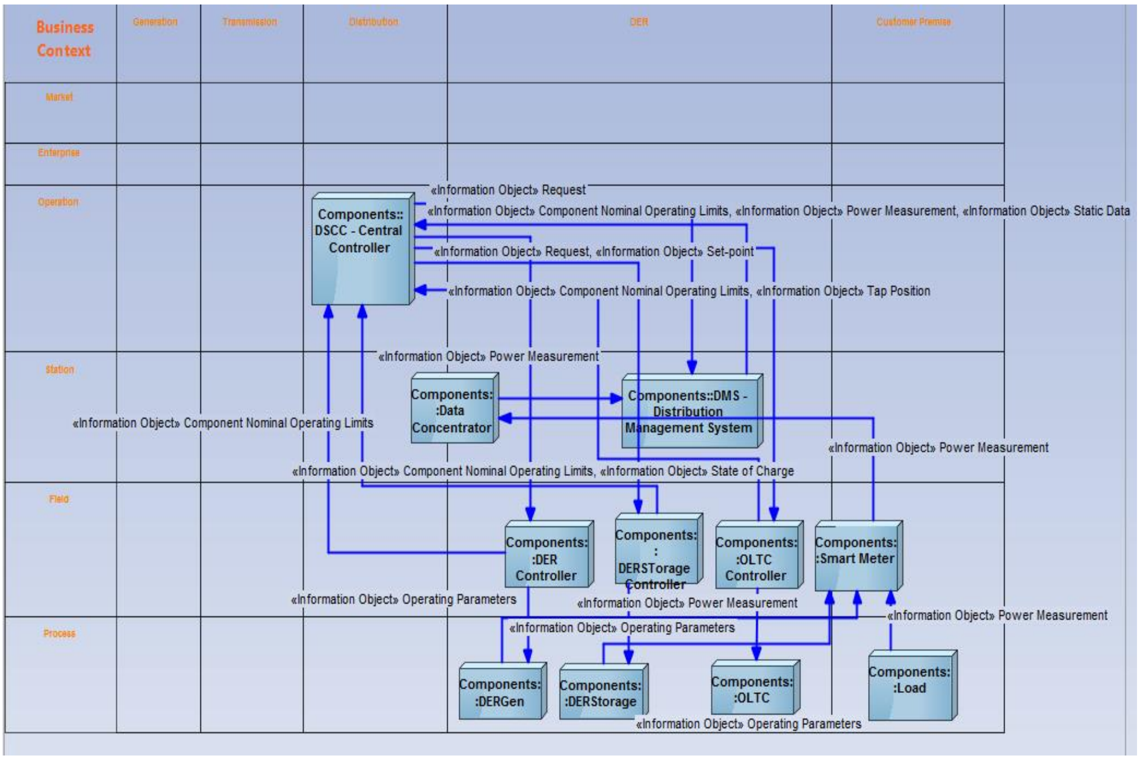


Setpoint Dispatch

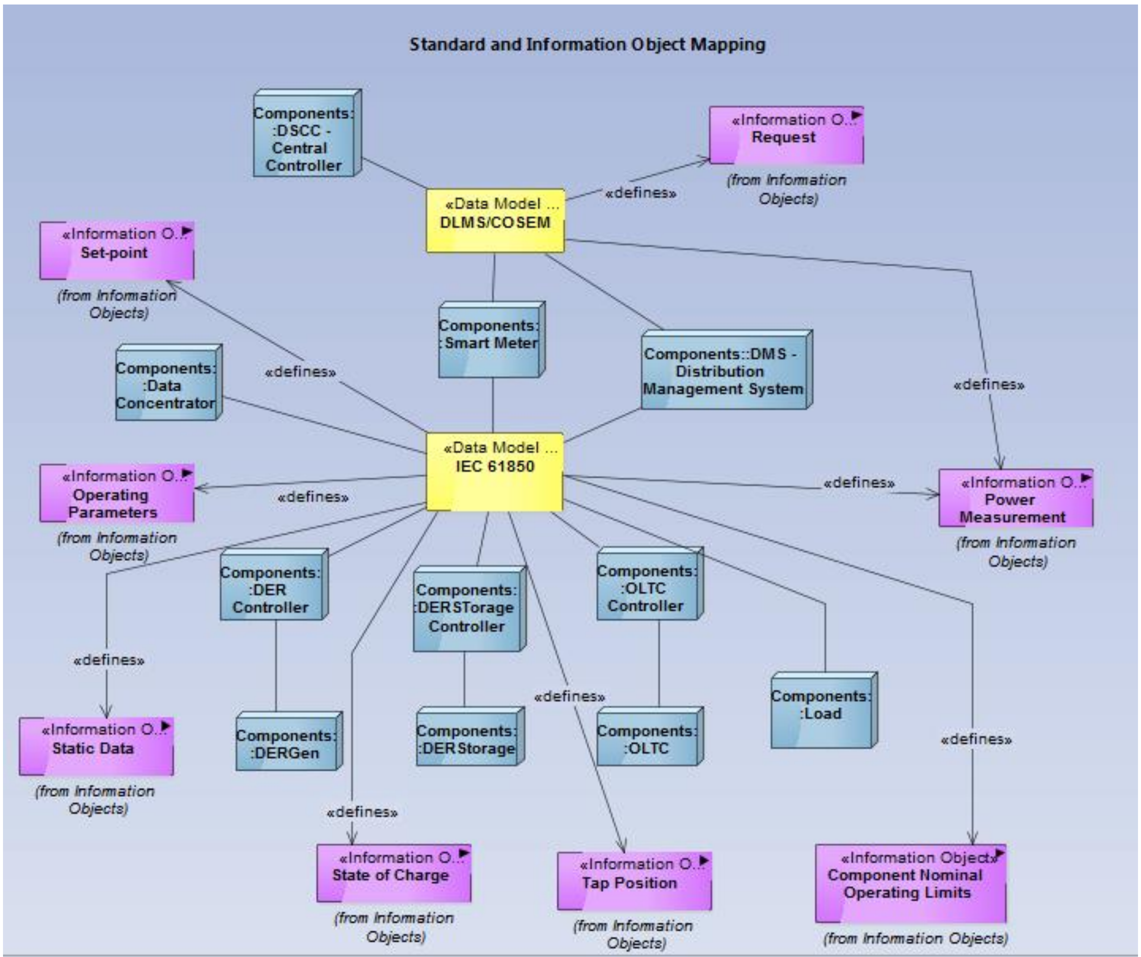
Information Layer

Based on paragraphs 4.2 and 5 the diagrams of information layer are presented below. The data models for the exchange of information objects are defined.

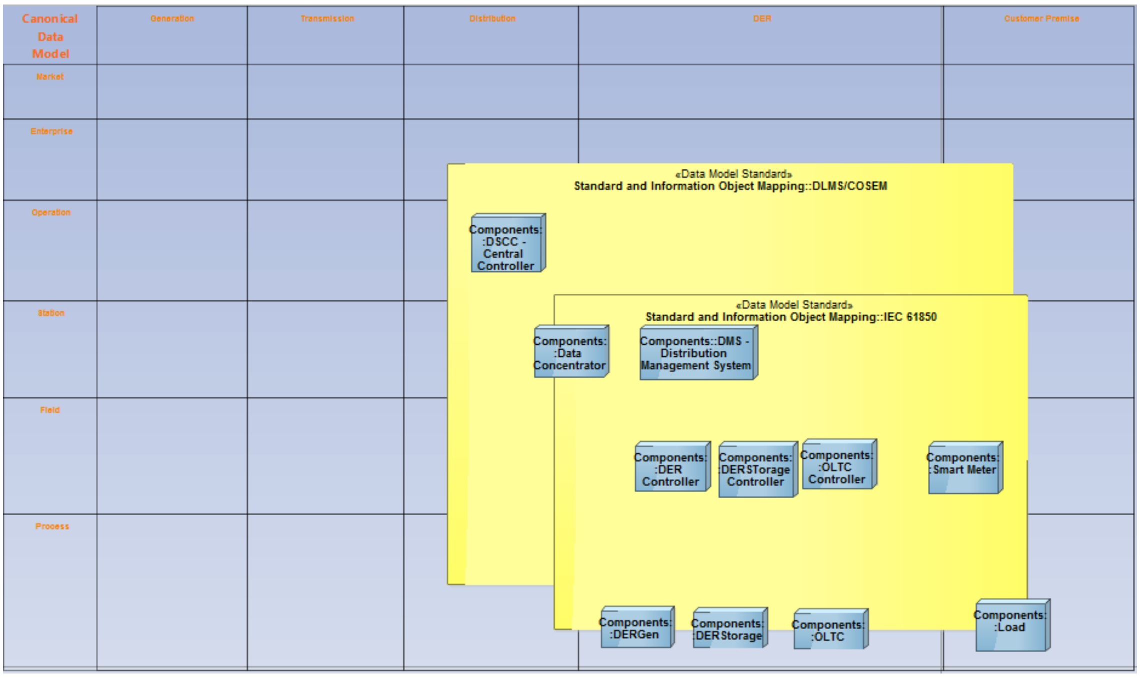
Information ex-changed ID	***Information Object***	***Relative Data Model***
1	Request	IEC 61850 - DLMS/COSEM
2	Power measurement	IEC 61850 - DLMS/COSEM
3	Set-point	IEC 61850
4	Operating parameters	IEC 61850
5	Static data	IEC 61850
6	State of charge	IEC 61850
7	Tap position	IEC 61850
8	Component nominal operating limits	IEC 61850



Information objects exchanged



Information objects mapped to standards

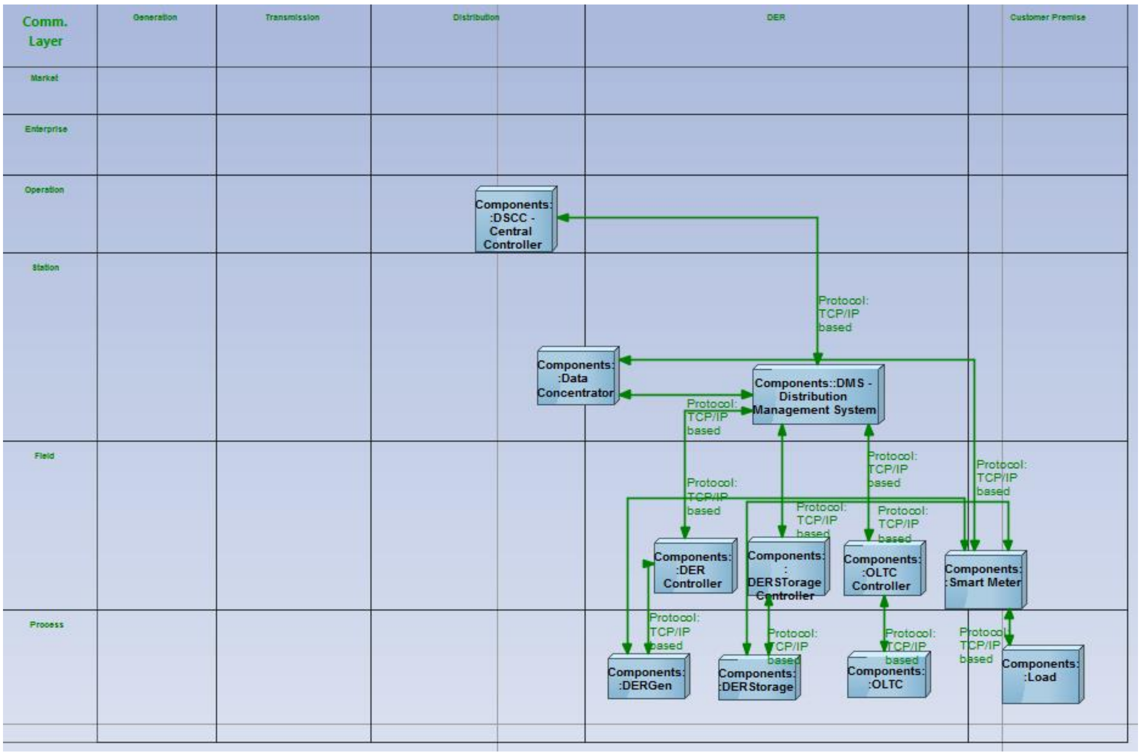


Information layer

Communication Layer

The protocols used for the communication between the components are presented in the diagram below.

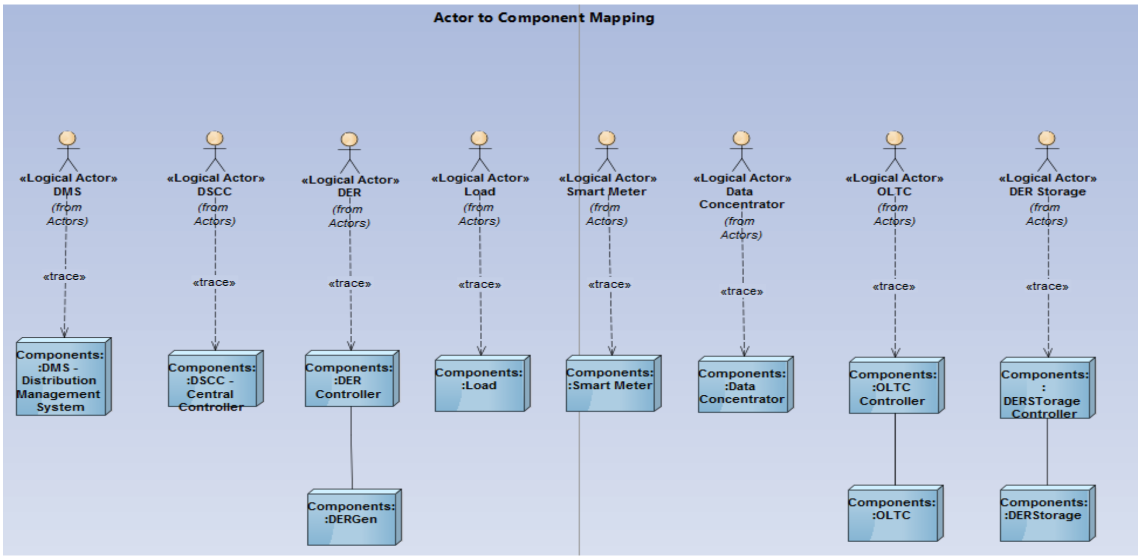
Communication Links		
Communication Link	***Components***	***Protocol***
1	DSCC - DMS	TCP/IP
2	Data Concentrator - Smart Meter	TCP/IP
3	Data Concentrator - DMS	TCP/IP
4	DMS - DER Controller	TCP/IP
5	DMS -OLTC Controller	TCP/IP
6	DMS -DER Storage Controller	TCP/IP
7	DER Controller - DER Gen	TCP/IP
8	DER Storage Controller - DER Storage	TCP/IP
9	OLTC Controller - OLTC	TCP/IP
10	Smart Meter - Load	TCP/IP
11	Smart Meter - DER Gen	TCP/IP
12	Smart Meter - DER Storage	TCP/IP



Communication layer

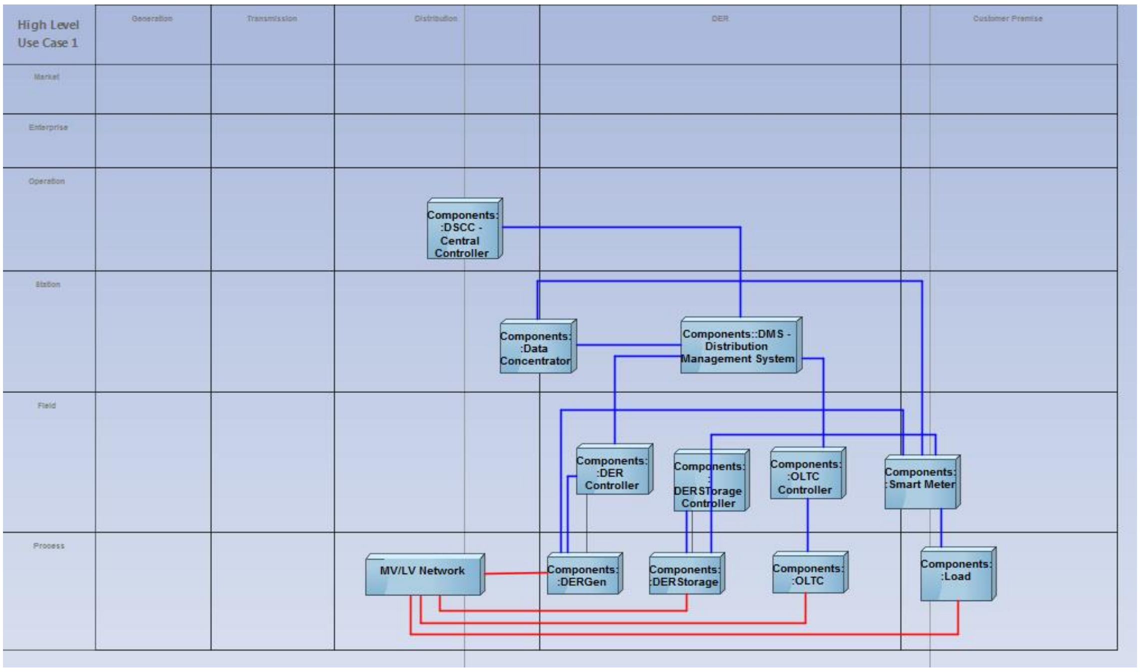
Component Layer

In order to actually implement the functionalities described in coordinated voltage control, the mapping of "logical actors" with the real components is necessary. The diagram below presents the relation between them.



Mapping of actors to real components

The implementation of the CVC is presented in the diagram below, where the connections (electric or ICT) among components are depicted. The position of each component in the diagram is defined by their role in the energy conversion chain (horizontally) and in the power system management (vertically).



Component layer

9.5 A detailed analysis of current practices at ERIGrid Research Infrastructures

9.5.1 Questionnaire template

Questionnaire on testing procedure used in the ERIGrid consortium

Name of institution:

Names of contributors:

Questionnaire	Application
Purpose of investigation	
Object of investigation / test object *(components, procedures, configurations, architecture, algorithm, ...)*	
Type of investigation / test / experiment *(Laboratory Experiment, (co-)simulation, HIL, field test, ...)*	
Properties to be tested / test criteria	
Performance indicators to assess properties / test criteria *(How do you evaluate these properties?* *Are the results compared with standards or requirements?)*	
Test setup *(Which objects are involved? Which preparation steps are conducted?)*	
Controllable input parameters and range *(Which input parameters are set for each test?* *Which parameter values are considered, e.g. intervals or discrete values?)*	
Uncontrollable input parameters *(Are input parameters considered that cannot be controlled, e.g. outdoor temperature or global radiation? Which parameters values are considered?)*	
Experimental design *(Is the test repeated with varied input parameter settings?* *[How] are input parameter settings varied?* *Which combinations of inputs do you consider?* *e.g.* *- all possible combinations* *- random combinations* *- combinations selected by hand* *- a systematic plan (design) of combinations* *- ...)*	
Test procedure *(step-by-step description of the test procedure.* *Is a standard for testing used? If yes, which one?)*	

Testing tools / libraries used and corresponding purpose	
Data exchange between components (interfaces, protocols, standards)	
Data storage / filing of experiment results	
Data evaluation tools / libraries used *(With which tools are the results processed?)*	
Sources of errors / uncertainty *(What kind of errors or uncertainties might occur during testing? E.g. uncertainties resulting from model concepts, numerical methods, measurement errors, ...)*	
Propagation of errors / uncertainty *(Do you consider uncertainty that propagates through your test setup? If yes, which methods do you use to quantify it?)*	
Example cases (projects, papers, …)	

The main results of the working groups are summarized in the following.

9.5.2 Object of Investigation

Gives an overview of the clusters that were identified among the objects of investigation defined in questionnaire.

Table 9.5: Overview of objects of investigation

Cluster	SubCluster	Description
Abstract Objects	Architecture	Topology of the grid, integration of an entity or concept to the grid.
	Protocol	Communication protocols, standards and norms.
	Modelling	Methodologies for modeling, dynamic models, Simulation setup and modeling error.
	Algorithm	Control and protection strategies, integration of advanced controllers.
Physical Objects	Grid	High, Medium, Low Voltage Networks and all grid equipment (grid analyzer, protections, transformer, circuit breaker, etc.).
	DER	Distributed Energy Sources (PV, ESS, Wind, Generators, Heat Storage Systems, etc.), integration of DER to the grid.
	ESS	Energy Storage Systems (batteries, flywheels, super capacitors, etc.)
	Power Electronic Device	Power conversion systems (i.e. inverter, converter, EV charger)
	EV	Electric Vehicles, integration of EV to the grid.
--	Other	Objects that do not fall into any of the aforementioned classes.

Figure 9-1 shows how the categories are broken down to sub-clusters. The two most interesting objects appear to be: Power Electronic Devices (17%) and Algorithms (15%). Others are distributed quite evenly. The physical components draw a lot of attention and make up 51% of the objects of investigation. 37% of the objects of investigation were categorized as abstract objects. 12% were categorized as other objects.

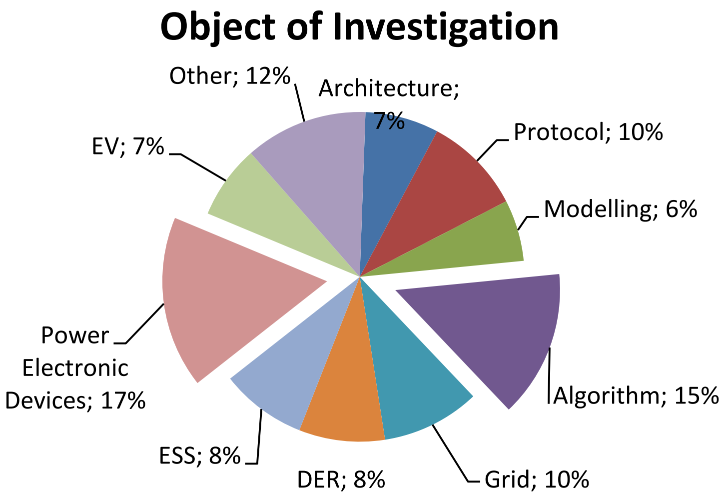


Figure 9:1: Objects of Investigation in more detail

In a further investigation, the conceptual model of NIST[10] for smart grids has been used as a basis, more specifically the model of the Distribution Grid (see Figure 5-nist-distribution-grid). The objective has been to highlight common NIST conceptual model domains of interest of the proposed test-cases in the form of a heatmap.

To this end, the objects of the test-cases were mapped into the corresponding domains of the NIST conceptual model. From the obtained table, a heatmap has been generated and superposed over the distribution model as shown in Figure 9.2.

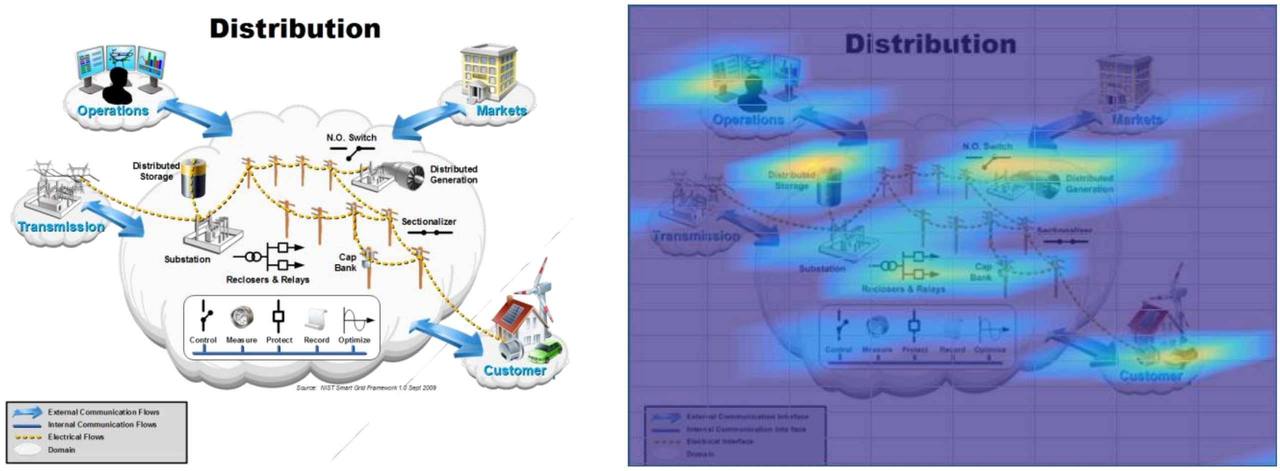


Figure 9.2: NIST model of the distribution grid and heatmap of domains

Huge attention was drawn to the aspect of Operations (Algorithm), Customer (Integration of electric vehicles (EV)), Distributed Storage (ESS - Battery), and distributed generation (photovoltaic (PV), wind). Little to none interest on Transmission and Market aspects.

[10] National Institute of Standardization and Technologies (U.S)

A large part of the test-cases in the domain of Operations is about algorithms for the integration of EVs and distributed energy resources (DER) to the grid. Also, most of the Power Electronic Devices and Grid Component test-cases are about inverter/converter serving for integration of PV–DER.

9.5.3 Purpose of Investigation

The field for "purpose of investigation" in the questionnaire has been used by the ERIGrid consortium to describe specific testing procedures, i.e. the aim or goal of the test or experiment. Due to the diversity of these procedures, the objective of the working group has been to come up with a straightforward and concise list of clusters that put any possible testing procedure under one of the proposed categories with common properties. The proposed clusters ought to cover a wide range of domains and applications so that they will meet the ERIGrid objectives.

The finally selected approach aims at reflecting the system under test, and the methodology that the testing procedure describes. The description of the two aspects was deemed necessary by the group because the purpose of investigation should succinctly give the direct overview of the testing procedure in general and, in addition, to facilitate the determination of a holistic approach in the frame of ERIGrid. Also, the selected clusters ought to represent and reflect approaches of the other working groups as well, such as Test Criteria and Design, as much as possible. In addition, for simplicity's sake it was considered of high importance to have a final set of less than ten clusters with descriptive naming that, despite the fact that terminology is an issue that calls for a holistic approach, would be consistent with the project goal. The following table gives an overview of the selected clusters and justification for the specific selection.

Table 9.6: Selected clusters with regard to the purpose of investigation

Cluster	Description	Remarks
Verification of service provision, functional behaviour and conflict analysis in system-integrated approach	Components and algorithms are tested in integrated approaches so as to assess the overall system behaviour, verify the provision of requested service, functional behaviour and identify potential conflicts	Devices, algorithms and methods can be tested with regard to their proper functionality and impact on a system, emphasizing the overall behaviour and the capability of the component to provide the service/functionality that is required, and not the exact performance of the component or method
Performance evaluation of algorithm or equipment	Components (devices, algorithms) are evaluated individually in terms of performance (efficiency, accuracy)	Several testing procedures regard the assessment of components (devices, equipment, algorithms) performance in terms of efficiency and accuracy, based on standardised or other reference procedures.
(Sub)Scenario assessment and validation	Overview of power systems performance under various scenarios and assumptions	This type of testing procedure refers to power system evaluation of performance under general scenarios, assumptions and approaches, e.g. impact of high RES penetration on distribution grids.
Performance and response time of protection equipment	Protection equipment is evaluated in terms of performance and response time	Although protection equipment might well be considered as part of the performance evaluation, particularities and special requirements make a separate approach necessary in terms of PoI
Compatibility and interoperability of ICT components	Testing procedures that regard compliance with standards, compatibility and other interoperability issues of ICT components	Several testing procedures regard compatibility and interoperability of components, compliance with ICT standards, etc.

Cluster	Description	Remarks
Cyber Security of ICT	Testing procedures that regard security of ICT	There is one testing procedure concerned with vulnerability of ICT equipment. For obvious reasons, this purpose is worth as separate clustering in its own right

9.5.4 Test Criteria

The test criteria summarize the properties to be tested and the performance indicators with which these can be tested. Thus, this working group chose a classification scheme of the following design: first the information given in the columns "Properties to be tested" and "Performance indicators to assess properties" of the questionnaires would be used to classify test cases regarding their *a)* properties to be tested and *b)* the domain of the test.

The combination of both classes would afterwards be used to identify a cell in a matrix made up by these two dimensions. The addition of a 'domain' dimension was done in a top-down fashion, motivated by the anticipation of different test characteristics depending on the kind of the system under test. The following table shows the properties found (characterization and validation & testing) including a more detailed definition.

Table 9.7: Overview of properties classification

Characterization	Behaviour characterization	Model identification (white-box, grey-box, or black-box)
	Performance characterization	Given a performance criterion quantify characterize the performance of the test object
Validation and testing	Test for undesired behaviour	The test object should NOT show certain behaviour given certain test conditions. An example would be that a inverter component should not show oscillations in presence of interference from another inverter in electrical vicinity.
	Testing against one or more characteristic(s)	Functionality of the test object has to show behaviour according to a predefined characteristic (e.g., datasheet)
	Testing concordance with standard	The behaviour of the test object must comply with a specific standard. This is explicitly mentioned in the questionnaire
	Validation of accuracy/ functionality against a simulation/literature/theory	The test object should display behaviour according to a validated model or calculated characteristic.
	Qualitative behaviour	The test object has to display particular functionality. This functionality is not necessarily quantifiable, nor defined in standards.

The second clustering was for domains and their variables (e.g. electrical, thermal, ICT) distinguishing between single domain and multi-domain. The following summarizes the results:

- Single domain:
  - physical-electric, i.e. the object under test properties to be investigated are in electric domain
  - physical-other, i.e. the object under test properties in other physical domain (e.g. thermal)
  - ICT, i.e. the object under test properties only in ICT domain
- Multi-domain

- energy-electric, e.g. conversion unit focus (e.g. PV conversion efficiency, ...)
- more than two physical domains, e.g. thermal, light, electricity
- ICT-electric, i.e. ICT and electricity domain are evaluated jointly
- ICT-other, i.e. ICT and other physical domain are evaluated jointly.

9.5.5 Test Design

The questionnaire summary was reviewed in order to establish commonality within its 'test procedure' field giving the steps conducted within a test. This was the most direct piece of information related to the test design. It was also evident that other fields in the questionnaire have an impact on the test design. These are "test inputs" and "purpose of test".

Furthermore, it was clear that the there was a common process or flow for all tests that can be modified based on information in the aforementioned fields. A generic test flow is depicted in Figure 9.3. DUT, SUT and FUT in the figure refer to device, system and function under test respectively.

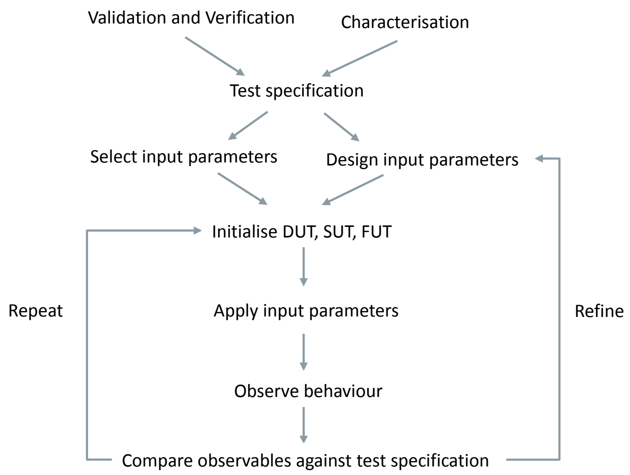


Figure 9.3: Generic test flow:
DUT - device under test, SUT - system under test and FUT - function under test

Only two types of tests can be discerned from the test cases:

- Validation and verification tests: these are tests for which the expected outcome is known and the test inputs are predetermined prior to the test.
- Characterization tests: these tests are conducted to investigate the response of the system for different inputs. The inputs in this case can be modified depending on the outcome of previous test iterations. Moreover, test inputs can be changed over the duration of a test or multiple test iterations based on the type of test.

The two key parameters that influence the test design (that is test inputs and purpose of test) lend themselves to the creation of a matrix where test cases can be readily mapped. However, further refinement of the criteria used for clustering was needed to ensure that no information is lost. In order to achieve this refinement, test flows were created for sample test cases to determine the common and differentiating aspects of each case. It was clear that the main variations were related to the test inputs, specifically in how they are devised and how they are modified between test iterations.

The main variations of the test input parameters based on the collection of test cases are as follows:

- Systematic/factorial: test inputs are varied in a systematic/rigorous fashion which maximizes coverage in terms of stimulating the behaviour of the system under test.
- Repeating: test inputs are modified based on the outcome of previous test iterations.

- Discrete/manual: test inputs, while may depend on previous test iterations, the variation of the inputs is not necessarily governed by a formal methodology.

Table 9.8 below summarises the main test flows and the test input variations as discussed above.

In summary, three main clusters have been identified based on the variations in input parameters. These can be split further into six clusters depending on the type of test (i.e. validation and verification or characterization).

Table 9.8: Overview of main test flows and test input variations

		Purpose of test	
		Validation and verification	**Characterisation**
Input parameter variation	**Systematic/factorial**	Validation and Verification Objectives elicitation/standard test requirements Select input parameters Initialise DUT, SUT Next test iteration Systematic/factorial inputs Observe behaviour Compare observables against test specification (1a)	Characterisation Define test envelope Design input parameters Initialise DUT, SUT Next test iteration Systematic/factorial inputs Observe behaviour (1b)
	Repeating	Validation and Verification Objectives elicitation/standard test requirements Next test iteration Select input parameters Initialise DUT, SUT Test input Observe behaviour Compare observables against test specification (2a)	Characterisation Define properties under investigation Next test iteration Design input parameters Initialise DUT, SUT Test input Observe behaviour (2b)
	Discrete/Manual	Validation and Verification Objectives elicitation/standard test requirements Select input parameters Initialise DUT, SUT Discrete manual input Observe behaviour Compare observables against test specification (3a)	Characterisation Define test observables Design input parameters Initialise DUT, SUT Discrete manual inputs Observe behaviour (3b)

9.5.6 Test Setup

Clustering has been initially performed identifying which columns in the questionnaire were of particular interest for "test-setup" clustering, i.e. giving insight in the objects involved and necessary preparation steps. In particular columns "Test setup", "Type of Investigation", "Object of Investigation" and "Testing Tools" have been considered.

The clustering has been conducted such that, for each procedure, a "YES/NO" classification is possible. Table 9.9 gives an overview of the resulting clusters. The identified clusters try to cover all peculiar aspects in Test Setup. In particular a distinction has been made between HW setup (involving also In-field test and the adoption of Real-Scale components) and SW setup that does not use any dedicated HW components (not considering simulation HW/SW platforms). More than 90% of the analysed testing procedures are characterized by HW Setup (30 of 32), about 85% test Real scale Components and 50% adopt grid and components simulators/emulators. These numbers are very promising regarding ERIGrid infrastructure capabilities for JRA3 and JRA4 activities. However, a limited number of procedures adopting Real Time and Hardware in the Loop methodologies (only 25%) have been observed.

Table 9.9: Overview of clusters regarding test setup

Cluster	Description
Hardware Set up	Test Set Up is based on Hardware installation of EUT and the test environment. (The HW set up can either be implemented in a controlled laboratory environment or in the field, using real components)
Software Set up	Test Set Up is based on modelling of the equipment or system under test and the simulation of the environment. No dedicated Hardware components are used.
In field Test	Hardware set up is not limited to Laboratory level but extends to real grid applications (as Real MV/LV networks, microgrids, etc.)
MV test capability	Test set-up allows test also in MV (HW or simulated)
Hardware and software integration	An integration of HW and SW is used (either off-line or in real time,). SW tools adopted to manage HW installation are not considered
Real Time and HIL	Real time integration (at different time domain) is used (including on-line management systems)
HW Grid and components simulator/emulator	Test set up foreseen the use of HW Grid simulator and emulator, Load and components (e.g. as Power electronic simulator/emulator)
Real scale components	Test set up allows to perform testing on real (full) scale physical components
Interoperability and communication	Specific test set up is used to test interoperability and communication (not considered ICT tools needed to manage laboratory infrastructures)
Automated test	Test are performed in an automated way using HW/SW automation tools to control the test environment as well as the EUT/SUT
Co-simulation or multi-domain simulation test	Tests are performed using a co-simulation or multi-domain simulation approach, e.g. by simulating the electrical domain and the communication/control domain or integrated electrical/thermal/gas domains test

9.5.7 Possible interconnections between tests

The information on interconnections of tests and involved domains are important to show the potential of coupling tests from different RIs within a holistic set up. To this end, questions have been gathered regarding which interconnections there are within and between domains and how data can be exchanged.

Based on the answers, the types of exchanged data have been grouped. This data has been input, output, or both to the described tests. This is especially interesting for the discussion on which information of input and output parameters will be needed apart from the variability attributes and target metrics. The groups that have been found are listed in the following, together with some examples:

- Electricity measurements (e.g. voltage, current)
- Network Topology and Parameters (e.g. switch or breaker states, grid impedances)
- Environmental Conditions (e.g. Solar irradiance, Wind speed)
- Market/Operation/Planning (e.g. tariffs, policies)
- Thermal (e.g. load, fuel consumption)
- Miscellaneous (e.g. Firmware updates, Energy Conversion)
- Meta-data (e.g. Alarms/Event signals, Time/date stamps)

In general, the data flows in RIs can be classified into two types:

- Real time communication (including main alarm and events)
- Offline communication (data may be saved into models or logged)

All the partners possess a means for data storage and logging. The level of detail and the frequency/resolution varies as well as the chosen tools.

The nature of exchanged data also varies from experiment to experiment.

It is unclear which data models are employed at local RIs. Manual harmonization (bilateral communication) may be necessary for bridging a certain gap at this level. A more systematic approach is to provide a common information model, with possible gateway for partners.

The used protocols create a wide spectrum that covers the communication system of a grid, from physical layer to information layer. It demonstrates the interest of possible collaboration among partners with different focuses in infrastructure.
Common points are:

- Many partners use protocols based on TCP/IP stack.
- OPC (UA and classic) is commonly used for SCADA applications.

Differences are:

- The used protocols reflect strongly the difference in research interests of partners. While some RIs focus on electrical domain with only protocols up to SCADA and substation layers (Modbus, Profibus, IEC 61850, OPC), some RIs work mainly in ICT domain with many protocols at communication layers (PRIME, IEC 62056, XML-RPC).
- Some partners use secured protocols (DNP3, IEC 61850) while others do not.

Possible Gaps are:

- While CIM and IEC 61850 possess their own information models, the classic protocols do not. An information model on those protocols, if necessary, must be created manually.
- Security issues among different protocols.

- Various protocols are used between SCADA layer and physical layer (Modbus, Profibus, IEC 61850, etc).
- Time synchronization and signal harmonization in the communication among partners.

Regarding the gaps above, physical connection is probably not feasible, however a “virtual” connection (Information and Application layers) might be possible. OPC UA seems to be a suitable candidate because:

- OPC is widely used.
- Even though OPC UA differs from classic OPC, bridging is possible.
- No necessity in changing actual infrastructure.
- OPC UA supports communication beyond LAN network, which classical OPC does not.
- The data model of OPC UA is extensible and is adaptable to lower/higher level information model (IEC 61850 and CIM, for instance)

9.6 Test Description Templates

9.6.1 Holistic Test Case Template

Name of the test case	Name
Narrative “a storyline summarizing motivation, scope and purpose of the test case.”	*What is the subject of the test and why is the purpose of the test?*
System under Test (*SuT*): “a (specific) system configuration that includes all relevant properties, interactions and behaviours (closed loop I/O and electrical coupling), that are required for evaluating an OuI as specified by the test criteria. “ A list of systems, subsystems, components included in the test case or test setup.	*What is the test system & the test system boundary? What is the system context and which interactions between your object under investigation and the surrounding system are relevant? What are the “external” interactions across the system boundary?* *If possible, provide an illustration and utilize a formal (referential) system specification?*
Object under Investigation (*OuI*) "the component(s) (1..n) that are to be characterized or validated”	*Which is the actual subject of this test case? Identify the sub-system(s) or component(s) that is/are in focus for this test. It may be listed above or a part of the systems listed above.*
Domain under Investigation (*DuI*): “Identifies the relevant domains or sub-domains of test parameters and connectivity.”	*Which interactions are part of the test case? Which domains of expertise needs to be included/emulated in a potential test setup? In a multi-domain system, not all interactions need to be reflected in a test; identify the domains and/or sub-domains that are relevant for this test case.*
Functions under Test (*FuT*) “the functions relevant to the operation of the system under test, as referenced by use cases”	*Which use cases apply to this test case or which system functions are required for an operational FuI to be investigated? List all functions required to be operational in the final test setup.*
Function(s) under Investigation (*FuI*) “the referenced specification of a function realized (operationalized) by the object under investigation”	*The function or sub-function that is operational in the OuI and subject to testing.*
Purpose of Investigation *(PoI)* “a formulation of the relevant interpretations of the test purpose (e.g. in terms of Characterization, Verification, or Validation)”	*What information will be gained by a successfully carried out test? What is the objective of this evaluation? Use keywords such as* Characterization, Verification, *or* Validation, *as well as reference to properties of the OuI or FuI.*
Test criteria: “the measures of satisfaction that a need to be evaluated for a given test to be considered successful.” A formalization of the purpose of investigation wrt. SuT and FuT attributes.	*(this field can used for explanation on how the PoI is broken down; or be left empty as the criteria are formalized in terms of the quantitative measures formulated below)*

Name of the test case		Name
	target metrics (criteria) A numbered list of measures to qualify (quantify) each identified Purpose of Investigation	*Based on the PoI, formulate the central quantities which should be calculated and evaluated to determine the test outcome. What should be measured, and with what should it be compared?*
	variability attributes (test factors): identification of the sets of attributes (controllable or uncontrollable parameters) and qualification of the required variability; includes reference to purpose of investigation.	*Which system (input, state) parameters should we varied in order to disturb the OuI?* *What values should these parameters assume?* *What kind of faults should the system be subjected to?*
	quality attributes (thresholds): with reference to purpose of investigation and/or target metrics, the threshold level required to pass a test or the certainty/precision level (e.g. probabilistic measure) required for the quality of a characterization	*How good should the target metrics be quantified in order to decide the test outcome? This field identifies the stopping criteria of a test in terms of constraints or thresholds of the target metrics (e.g. actual acceptable minimum or maximum values). In case of characterization tests, here also the required range and statistical quality of the test outcome can be specified.*

9.6.2 Test Specification Template

Title	Definition
Ref. Holistic test case	
Test System (also graphical)	*Graphical and textual description of the system under investigation and its components including interfaces between test setup and Object under investigation and type of those interfaces (e.g. electrical)*
Target measures	*Specification of the target metrics that will be derived from measured parameters in order to evaluate the test objectives. Which variables will be quantified by the test? (formula and explanation)*
Input and output parameters	*List of inputs for the system under test relevant to the object under investigation, inputs relevant to the object under investigation itself and outputs / measured parameters divided into:* • *'Controllable input parameters'* • *'Uncontrollable input parameters'* • *'Measured parameters'*
Test Design	*The choice of mapping between required testing target and available test parameters, in terms of test sequence, decision criteria and controlled parameters. Textual or graphical description of the sequence of steps carried out during the test including parameter ranges and variation of input parameter.*
Initial system state	*Description of conditions that are prerequisites to actually run the test and initial choices of parameters.*
Evolution of system state and test signals	*Quantitative characterization of the temporal evolution of test events and evolution of the relevant test parameters, as adjustable by the input parameters (e.g. opening breakers after a certain amount of seconds); incl. variability attributes*
Other parameters	*Information of data that should be tracked apart from the input and output parameters and system state, test signals*
Storage of data	*In which format are the parameters stored?*
Temporal resolution	*Discrete or continuous simulation and (if applicable) resolution of the discrete time steps*
Source of uncertainty	*In order to evaluate the quality of the test, the possible sources of uncertainties are given in how they can be quantified.*
Suspension criteria / Stopping criteria	*Under which conditions are the test results not valid or the test is interrupted*

9.6.3 Experiment Specification Template

Title	Definition
Ref. Test Spec.	*Reference to test specification document.*
Research Infrastructure	*Specify the RI where the experiment is carried out.*
Experiment Realisation	*The setup can be realised in different ways (e.g. simulation, hardware,…): give a brief description of the realization.*
Experiment Setup (concrete lab equipment)	*Graphical and textual description of the concrete lab equipment and interconnections*
Experimental Design and Justification	*For all parameters give a reason why it has been chosen that way* • *concrete values, sequences of values of "variability attributes" and* • *concrete combinations of different variability attributes* • *number of repetitions for each combination*
Precision of equipment	*For the components of the lab equipment the precision is given such that the experiment's uncertainty can be derived.*
Uncertainty measurement	*Based on the precision of equipment of the lab instrument and of measurement algorithms, the parameters to model the measured quantities' errors are provided it is specified how experiment's uncertainty can actually be measured.*

9.7 Mini-Tutorial: Distinguishing Test Case, Use Case, and System Configuration

- **Examples:**
  1. Real-time optimal coordinated voltage control in distribution networks
  2. Agent-based control system for controlling CO2 emissions in virtual power plants
  3. Fault ride-through compliance of offshore wind power plants connected through HVDC based on voltage sourced converter technology

  Which is which?

- **Use case:** *Specification* of a set of *actions performed by a system*, which yields an observable result that is, typically, of *value* for one or more *actors* or other stakeholders of the system.
  - *"actions performed"* - action typically the first word
  - "by a *system"* – refer to a system (see below)
  - Often also the objective or *value* and the respective *actor* is stated

- A **Test case** provides a *set of conditions* under which a test can determine whether or how well a system, component or one of its aspects is working given its expected function.
  - A test needs to evaluate *conditions "how well"*
  - Of a system in its *operational context* ("working")

- **System -** Set of interrelated elements considered in a defined *context* as a whole and separated from their *environment.*

  Which is which?
  1. – *Use Case*; 3. *Test Case*
  2. – *unclear* – "system" named after use case objective

TRY IT OUT

"Validation of aggregator performance in delivering secondary control ancillary service"

- [ ] Test Case
- [ ] Use Case
- [ ] System / Component

"Aggregator Service delivery"

- [ ] Test Case
- [ ] Use Case
- [ ] System / Component
- [ ] unclear

"Aggregator secondary control ancillary service with Distributed Resources for Distribution system operators"

- [ ] Test Case
- [ ] Use Case
- [ ] System / Component
- [ ] unclear

"Aggregator for Resources connected to the Distribution Network"

- [ ] Test Case
- [ ] Use Case
- [ ] System / Component

"Validation of aggregator performance in delivering secondary control ancillary service"

- **[x] Test Case**
  keyword: "validation…" with corresponding use case "delivery of secondary control service"
- Not: [ ] Use Case [ ] System / Component

"Aggregator Service delivery"

- *Not clear if:*
  - *[ ] Test Case – is it a test at all?;*
  - *[ ] Use Case – what service?;*
- NOT: *[ ] System / Component*
  – more than system (noun/object); "service delivery" is an action.
- **[x] unclear**

"Aggregator secondary control ancillary service with Distributed Resources for Distribution system operators"

- [ ] Test Case – *no evaluation objective mentioned.*
- **[x] Use Case** – all elements specified: action, system (performing the action), stakeholder
- [ ] System / Component; [ ] unclear

"Aggregator for Resources connected to the Distribution Network"

- [ ] Test Case – *no evaluation mentioned*
- [ ] Use Case - *no action specified*
- **[x] System / Component** – system with a purpose is identified

9.8 Glossary

Term	*Abbrev.*	*Description*	*Remarks*
Actor		Entity that communicates and interacts (source: IEC 62559)	Generic concept which includes systems as well as stakeholders or stakeholder types. Relevant to Use Case specification, not System Configuration.
Attribute		Identifiable association between a system configuration object and a value. An attribute is a property of object.	
Characterization (Test objective)		Type of test objective. Here, a set of measures is specified but no requirements are posed for the OuI for passing the test. A test is successful when a sufficient data can be collected. Examples: characterizing performance of a system; tests for developing a simulation model.	Aspect of Test Case specification; to be used in Test Objective.
Compliance		Accordance of the whole implementation with specified requirements or standards. However, some requirements in the specified standards may not be implemented.	[SOURCE: CEN-CENELEC-ETSI SG-CG Report on Interoperability CEN_9762_CLC_9624 – clause 12.1 Terms and definitions]
Component		Constituent part of a system which cannot be divided into smaller parts without losing its particular function for the purpose of investigation.	(Based on IEC 60050 (151), replacing "device" with "system"). In a system configuration, components cannot further be divided; connections are established between components.
Conformance		Accordance of the implementation of a product, process or service with all specified requirements or standards. Additional features to those in the requirements / standards may be included. All features of the standard/specification are implemented and in accordance, but some additional features are not covered by the standard / specification.	[SOURCE: CEN-CENELEC-ETSI SG-CG Report on Interoperability CEN_9762_CLC_9624 – clause 12.1 Terms and definitions]
Conformance Testing		The act of determining to what extent a single implementation conforms to the individual requirements of its base standard. An important condition in achieving interoperability is the correct implementation of the standards. This can be verified by conformance testing. Determines whether an implementation conforms to a profile as written in the PICS. The latter testing can be interoperability testing if profile covers the interoperability requirements additional to the conformance testing requirements of standards applied. Conformance testing is a prerequisite for interoperability testing.	[SOURCE: CEN-CENELEC-ETSI SG-CG Report on Interoperability CEN_9762_CLC_9624 – clause 12.1 Terms and definitions]
Connection Point (System Configuration)	CP	Logical concept in a System Configuration, required to establish a connection between two terminals from the same domain.	Logical CP is always present in a connection btw. Terminals. By convention, it is only drawn if more than 2 terminals are being connected.
Connectivity - Abstract Data	C-AD	Connectivity classification as: AD - Abstract Data, such as aggregated or stored field data or otherwise abstracted and data, such as configuration data: only highly processed information is transferred from/to this component/domain. Expresses an "about" relation rather than a concrete connection.	Connectivity attribute applicable to GSC. Informational / conceptual relation. Example: Asset Management information about physical components.

Term	Abbrev.	Description	Remarks
Connectivity - Direct Data	C-DD	Connectivity classification as: DD - Direct Data: direct field-related data for real-time control & decision purposes; e.g. as recorded in the field, is transferred from/to this component	Connectivity attribute applicable to GSC. Concrete communication connection.
Connectivity - Direct Physical	C-DP	Connectivity classification as: DP - direct physical coupling (intra-domain)	Connectivity attribute applicable to GSC. Concrete physical connection.
Connectivity - Indirect Physical	C-IP	Connectivity classification as: IP - indirect physical coupling (either mediated, e.g. by a power converter by other technique; also applicable to 'equivalenced' components)	Connectivity attribute applicable to GSC. Multiple physical domains involved - unspecific.
Connectivity - Stakeholder - Directive	C-S-D	(D)irective - Stakeholder directs Components or other Stakeholders	Connectivity in abstract role model.
Connectivity - Stakeholder - Informational	C-S-I	(I)nformational - S. acquires information from	Connectivity in abstract role model.
Connectivity - Stakeholder - Operator	C-S-OP	(OP)erates - S. Operates component	Connectivity in abstract role model.
Connectivity - Stakeholder - Owner	C-S-O	(O)wnership - S. owns component	Connectivity in abstract role model.
Connectivity - Stakeholder - Responsible	C-S-R	(R)esponsible - Stakeholder is responsible for Domain/Component	Connectivity in abstract role model.
Connectivity - Stakeholder - Transactive	C-S-T	(T)ransactive - S. executes transactions with respect to component/domain	Connectivity in abstract role model.
Connectivity (connection)		The link between two components in a system configuration. A connection is associated with a specific Domain.	In the SC methodology, a connection is established by association of two or more Terminals from the same domain with the same Connection Point.
Constraint		Limitations on Attribute values in a System Conf.	Part of System Configuration
Controllable Input Parameter	CTR-IP	Input parameters of a test system. Controllable parameters are distinguished and must be considered in the test design.	DoE concept.
Controller hardware in the loop	C-HIL	A hardware in the loop setup where the sensors and actuators of a control hardware, e.g. Protection equipment, are interfaced with in a real-time simulation.	
Description of Action	DoA	Project plan describing the intended actions and deliverables of a project.	Horizon 2020 terminology. Here typically refers to ERIGrid DoA.
Design of Experiments	DoE	A systematic method to determine the relationship between factors affecting a process and the output of that process.	Methodology applicable to the design and evaluation of experiments. Refers to mathematical framework. Related terms: (controllable, uncontrollable) Input Parameter, Output Parameter, Target Metric, Test System.
Domain		An area of knowledge or activity characterized by a set of concepts and terminology understood by practitioners in that area. Source: IEC 62559 (from ISO/IEC 19501:2005)	In a system configuration, domains represent a categorization of the connections between systems; a domain can be divided into sub-domains; domains interface with other domains via components.

Term	Abbrev.	Description	Remarks
Domain Type Hierarchy	DTH	A hierarchy of of Domains used to identify relations between different connection concepts.	Can be generic or test case specific. Aspect of GSC and SC. Can be used to trace the compatibility of terminals and identify the required additional specifications in various mapping steps (e.g. TC-GSC-->TS-SC)
Domain under Investigation	DuI	Identifies the relevant domains of test parameters and connectivity	Part of Test Case specification.
Experiment Setup	E-SC	Configuration of lab (RI) components as part of experiment specification.	cf. Experiment Specific System Configuration; also: "Test Setup"
Experiment Specific System Configuration	E-SC	Configuration of lab (RI) components as part of experiment specification.	Also called "Experiment Setup" or "test setup".
Function(s) under Investigation	FuI	The referenced specification of a function realized (operationalized) by the object under investigation.	The FuI are a subset of the FuT.
Function(s) under Test	FuT	The functions relevant to the operation of the system under test, as referenced by use cases.	The reference would typically be to a use case document; a preliminary identification of functions by function names and placement in a UC-GSC (e.g. In SGAM Plane) is typically acceptable in a Test Case.
Functional Requirement		(set of) function(s) the OuT or SuT must satisfy in order to pass the test. Functional Requirements are specified in Use cases.	
Generic System Configuration	GSC	Establishes the relevant types of systems, domains and connections relevant to a generic context (e.g. test case, use case, type of laboratory).	A generic system configuration (GSC) establishes the “semantics” and the types of concepts to be employed in a specify system configuration (SC).
Hardware in the loop	HIL	A test setup that combines a real-time simulated system with a physical component or system, where interfaces between physical and simulated systems enable closed loop interactions.	
Holistic Testing		The process and methodology for the testing of a system or component (treated as a distinct object) within its functional context. This context, or environment, is the encompassing and surrounding systems and subsystems stretching across domains such as electric power and ICT.	
Information and Communication Technology	ICT		Common acronym.
Input Parameter	IP	Input parameters of a test system. Controllable and uncontrollable parameters are distinguished and must be considered in the test design.	DoE concept.
Interchangeability		Interchangeability is the ability of two or more devices or components to be interchanged without making changes to other devices or components in the same system and without degradation in system performance.	[SOURCE: CEN-CENELEC-ETSI SG-CG Report on Interoperability CEN_9762_CLC_9624 – clause 12.1 Terms and definitions]
Interface		1. a shared boundary between two functional units, defined by various characteristics pertaining to the functions, physical signal exchanges, and other characteristics. 2. a hardware or software component that connects two or more other components for the purpose of passing information from one to the other	Source: 1. ISO/IEC 2382-1:1993, Information technology — Vocabulary — Part 1: Fundamental terms.01.01.38. 2. (ISO-IEC-IEEE 24765.2010)

Term	Abbrev.	Description	Remarks
Interoperability		Interoperability is the ability of two or more components, devices, networks, applications, systems or subsystems to exchange and use information for performing required functions.	
Interoperability testing		Interoperability testing should be performed to verify that communicating entities within a system are interoperable, i.e. they are able to exchange information in a semantically and syntactic correct way. During interoperability testing, entities are tested against peer entities known to be correct. (profiles)	[SOURCE: CEN-CENELEC-ETSI SG-CG Report on Interoperability CEN_9762_CLC_9624 – clause 12.1 Terms and definitions]
Key performance indicator	KPI	derived or directly observable indicator of quality, typically in form of a quantification of goal metrics.	
Lab System Configuration	L-SC	Lab configuration with components, including potential multiplicity and potential connectivity of lab components, but may have undefined connectivity.	also: Lab specification by lab entries in RI database; (using NA5.3 specific lab profiles)
Lab type Generic System configuration	L-GSC	Specification of Lab profiling information model; RI profile description model.	Can be used to define "templates" for (NA5.3) RI-specific lab profiles.
Narrative (Test Case)		A storyline summarizing motivation, scope and purpose of the test case.	Part of Test Case specification.
Object(s) under investigation	OuI	The object/component(s) (1..n) that are to be characterised, verified or validated by a test.	
Output Parameter	OP	Measurable or directly observable attributes of a Test System	DoE concept.
Parameter		A configurable Attribute of a component in a System Configuration.	Note: distinguish from DoE concept Input/output parameters.
Performance requirements		Quantification of a (set of) function(s) or property(ies) the OuT or SuT must satisfy in order to pass the test.	
Power hardware in the loop	P-HIL	A hardware in the loop setup where at least one of the bi-directional interfaces of a setup is to-scale electrical power.	
Purpose of Investigation	PoI	A list providing the relevant interpretations of the test purpose (test objective) in terms of Characterization, Verification, or Validation.	Part of Test Case.
Quality Attributes (thresholds)		With reference to purpose of investigation and/or target metrics, the threshold level required to pass a test or the certainty/precision level (e.g. probabilistic measure) required for the quality of a characterization	Aspect of Test Criteria specification.
Research Infrastructure	RI	Generic term that identifies a laboratory or other context or hosting infrastructure for test and experiments.	DoA term.
Scenario (DoA)		The compilation of System configuration, Use Cases, and holistic test cases in a shared context.	DoA term.
Scenario (high-level)		A quantification of demand, storage, grids etc. derived from a qualitative description of a future situation.	Based on Methodology for scenario-quantification (e-Highway 2050). ERIGrid usage: reference in System configuration.

Term	Abbrev.	Description	Remarks
Smart Grid Architecture Model	SGAM	High level conceptual model of the Smart Grid describing the main actors of the Smart Grid and their main interactions. Introduced in IEC 62357-16 Ed2	[SOURCE: CEN-CENELEC-ETSI SG-CG M490 Set of standards report CEN_9762_CLC_9624 Section 7.3 SGAM introduction] ftp://ftp.cencenelec.eu/EN/European Standardization/HotTopics/SmartGrids/SGCG_Standards_Report.pdf
Specific System Configuration	SC	Instance model of a system configuration. Refers to types and concepts defined in a Generic System configuration as well as to domain types defined in a domain type hierarchy.	A generic system configuration (GSC) establishes the “semantics” and the types of concepts to be employed in a specify system configuration (SC).
Sub-domain		An internal domain which is a part of a primary domain with more particular common concepts and terminology.	
System (generic)		Set of interrelated elements considered in a defined context as a whole and separated from their environment.	In a system configuration, a system represents a grouping of components, which may be divided into sub-systems; interfaces between systems a system.
System (use case)		A typical industry arrangement of components and systems, based on a single architecture, serving a specific set of use cases.	
System Configuration	SC	An assembly of (sub-)systems, components, connections, domains, and attributes. A system configuration can be generic (a domain model - GSC) or specific (a concrete instance - SC). Several forms of system configuration are distinguished.	System configuration or Systems configuration are used interchangeably. As description method, it provides a standardized way of representing systems that can be also multi-domain; related terms: Domain, Component and System, Connectivity, Constraints and Attributes.
System Configuration Concept Model		Defines the formal modeling concepts of a system configuration; Upper ontology for System Configuration.	specifies relations between and and attributes of Concepts: Domain, Component and System, Connectivity, Constraints, as well as a generic SystemConfigurationObject and SystemConfigurationContainer.
System Configuration Container	SCC	Data structure that holds a system configuration. Concept in System Configuration Concept Model.	
System Configuration Container Type	SCType	Depending on the specification context, this type Identifies the applicable rules for modeling and description.	i.e. UC-GSC, TC-GSC, TS-SC, E-SC, L-SC, L-GSC
System Configuration Diagram	SCD	Graphical (diagramatic) representation of a System Configuration, adhering to the System Configuration Concept Model for a specific SC Container Type. Includes representations of: Systems, Components, Terminals, Connection Points, Domains.	Related terms: Domain Type Hierarchy.
System under test	SuT	A (specific) system configuration that includes all relevant properties, interactions and behaviours (closed loop I/O and electrical coupling), that are required for evaluating an OuI as specified by the test criteria.	Part of Test Case specification.

Term	Abbrev.	Description	Remarks
Target Measures (/criteria)		A list of measures to qualify (quantify) each identified Purpose of Investigation in a Test Case.	Aspect of Test Case/Test Criteria specification.
Target Metrics	TM	A quantity that can be derived from test system parameters (input/output paprameters). Target metrics represent a quantification of the test criteria.	DoE concept; part of Test Specification.
Terminal (System Configuration)	T	A domain-specific interface point of a Component or System. Logical concept in a System Configuration.	based on CIM definition; compatible with "Interface"
Test Case	TC	A test case is a set of conditions under which a test can determine whether or how well a system, component or one of its aspects is working given its expected function.	With the attribute "Holistic" applied to a test case (holistic test case), the relationship to outcome of ERIGrid DoA Step1 is emphasized.
Test Case Generic System Configuration	TC-GSC	Generic SC applied to Test Case; defines test case relevant connection types, domains, range of multiplicities; identifies SuT	“class model” of SC
Test Criteria		The measures of satisfaction that need to be evaluated for a given test to be considered successful.	In a test case, also: "The measures of functionality or behaviour of a System under Test that are to be quantified."